ESC-TR-2007-063

Project Report
ATC-345

Uncorrelated Encounter Model of the National Airspace System Version 1.0

M.J. Kochenderfer
J.K. Kuchar
L.P. Espindle
J.D. Griffith

14 November 2008

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of Homeland Security under Air Force Contract FA8721-C-05-0002.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE
14 November 2008

2. REPORT TYPE
Project Report

3. DATES COVERED *(From - To)*

4. TITLE AND SUBTITLE

Uncorrelated Encounter Model of the National Airspace System, Version 1.0

5a. CONTRACT NUMBER
FA8721-05-C-0002

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Mykel J. Kochenderfer, James K. Kuchar, Leo P. Espindle, and J. Daniel Griffith

5d. PROJECT NUMBER
10082

5e. TASK NUMBER
1

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)

MIT Lincoln Laboratory
244 Wood Street
Lexington, MA 02420-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

PR-ATC-345

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
ESC-TR-2007-063

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Airspace encounter models, covering close encounter situations that may occur after standard separation assurance has been lost, are a critical component in the safety assessment of aviation procedures and collision avoidance systems. Of particular relevance to unmanned aircraft systems (UAS) is the potential for encountering general aviation aircraft that are flying under visual flight rules (VFR) and which may not be in contact with air traffic control. In response to the need to develop a model of these types of encounters, Lincoln Laboratory undertook an extensive radar data collection and modeling effort involving more than 120 sensors across the U.S. This report describes the structure and content of that encounter model. The model is based on the use of Bayesian networks to represent relationships between dynamic variables and to construct random aircraft trajectories that are statistically similar to those observed in the radar data. The result is a framework from which representative intruder trajectories can be generated and used in fast-time Monte-Carlo simulations to provide accurate estimates of collision risk.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	Same as report	110	**19b. TELEPHONE NUMBER** *(include area code)*

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Department of Homeland Security under Air Force Contract FA8721-05-C-0002.

This report may be reproduced to satisfy needs of U.S. Government agencies.

The ESC Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Gary Tutungian
Administrative Contracting Officer
Acquisition Enterprise Division

Massachusetts Institute of Technology
Lincoln Laboratory

Uncorrelated Encounter Model of the National Airspace System Version 1.0

M.J. Kochenderfer
J.K. Kuchar
L.P. Espindle
J.D. Griffith
Group 42

Project Report ATC-345

14 November 2008

Lexington Massachusetts

This page intentionally left blank.

ABSTRACT

Airspace encounter models, covering close encounter situations that may occur after standard separation assurance has been lost, are a critical component in the safety assessment of aviation procedures and collision avoidance systems. Of particular relevance to Unmanned Aircraft Systems (UAS) is the potential for encountering general aviation aircraft that are flying under Visual Flight Rules (VFR) and which may not be in contact with air traffic control. In response to the need to develop a model of these types of encounters, Lincoln Laboratory undertook an extensive radar data collection and modeling effort involving more than 120 sensors across the U.S. This report describes the structure and content of that encounter model. The model is based on the use of Bayesian networks to represent relationships between dynamic variables and to construct random aircraft trajectories that are statistically similar to those observed in the radar data. The result is a framework from which representative intruder trajectories can be generated and used in fast-time Monte Carlo simulations to provide accurate estimates of collision risk.

The model described in this report is one of three developed by Lincoln Laboratory. An encounter with an intruder that does not have a transponder, or between two aircraft using a Mode A code of 1200 (VFR), is uncorrelated in the sense that it is unlikely that there would be prior intervention by air traffic control. The uncorrelated model described in this report is based on transponder-equipped aircraft using a 1200 (VFR) Mode A code observed by radars across the U.S. As determined from a comparison against primary-only tracks, in addition to representing VFR-on-VFR encounters, this model is representative of encounters between a cooperative aircraft and conventional noncooperative aircraft similar to those that use a 1200 transponder code. A second uncorrelated model is also being developed for unconventional aircraft that have different flight characteristics than 1200-code aircraft. Finally, a correlated encounter model has been developed to represent situations in which it is likely that there would be air traffic control intervention prior to a close encounter. The correlated model applies to intruders that are using a discrete (non-1200) code.

Separate electronic files are available from Lincoln Laboratory that contain the statistical data required to generate and validate encounter trajectories. Details on how to interpret the data file and an example of how to randomly construct trajectories are provided in Appendices A and B, respectively. A Matlab software package is also available to generate random encounter trajectories based on the data tables.

A byproduct of the encounter modeling effort was the development of National aircraft track and traffic density databases. Example plots of traffic density data are provided in this report, but the complete track and density databases are not provided in electronic form due to their size and the complexity of processing specific locations, altitudes, and times.

This page intentionally left blank.

ACKNOWLEDGMENTS

This report is the result of research and development sponsored by the United States Department of Homeland Security Science and Technology Directorate, the United States Air Force 303rd Aeronautical Systems Wing (303 AESW/XRX), and the Department of Defense Unmanned Aircraft System Airspace Integration Joint Integrated Product Team.

The authors greatly appreciate the support and assistance provided by John Appleby, Kevin "Spanky" Kirsch, David Thomasson, Joseph Kielman, and Merv Leavitt from the Department of Homeland Security, and Maj. Luke Cropsey, Lt. Col. Kent Tiffany, Tony Salmonson, and Richard Graeff from the Air Force.

The authors appreciate the support from Hill Air Force Base, including Jeff Richardson, Steven Schimmelpfennig, Richard Whitlock, Lt. Han Saydam, Lt. Tanuxay Keooudom, James Evans, and TSgt. Christopher Cosper.

Finally, the authors would also like to thank Lincoln staff members Vito Cavallo, Ann Drumm, Matthew Edwards, Jeffrey Gertz, Robert Grappel, Garrett Harris, Jessica Olszta, John O'Rourke, Angela Sharer, and Steven Thompson for their contributions to the airspace encounter model effort.

This page intentionally left blank.

TABLE OF CONTENTS

This page intentionally left blank.

LIST OF ILLUSTRATIONS

LIST OF TABLES

This page intentionally left blank.

1. INTRODUCTION

One of the main challenges to integrating unmanned aircraft into the National Airspace System (NAS) is the development of systems that are able to sense and avoid local air traffic. If designed properly, these collision avoidance systems could provide an additional layer of protection that maintains or even enhances the current exceptional level of aviation safety. However, due to their safety-critical nature, rigorous assessment is required before sufficient confidence can exist to certify collision avoidance systems for operational use. Evaluations typically include flight tests, operational impact studies, and simulation of millions of traffic encounters with the goal of exploring the robustness of the collision avoidance system. Key to these simulations are so-called encounter models that describe the statistical makeup of the encounters in a way that represents what actually occurs in the airspace. Each encounter generated from the model specifies the initial positions and orientations of two aircraft and the nominal dynamic maneuvers (not affected by a collision avoidance system) that may occur leading up to the closest point of approach. What is produced, then, is the geometric, dynamic situation a collision avoidance system may be expected to resolve safely. Identical encounter situations are typically simulated with and without a collision avoidance system so that relative system benefits may be quantified. Knowledge of the rates at which encounter situations occur in the NAS can also be used to estimate the rate of near mid-air collisions per flight hour and thus arrive at an overall risk metric.

In these models, encounter situations are abstracted in the sense that there is no consideration of an explicit location or local airspace structure (e.g., airways, metering fixes, approach paths). Rather, the encounters represent what may be statistically expected to occur over the lifetime of a given system. If desired, a particular altitude layer or airspace class (e.g., Class E) can be specified and aircraft behavior will be representative of that particular region. Additionally, the flight path of one aircraft can be constrained to focus, for instance, on a particular departure profile or flight condition. It should also be noted that the models cover approximately 1 minute before closest point of approach and so are not appropriate for large-scale air traffic impact studies (e.g., to examine sector loading or conflict rates); the focus here is on events in which loss of separation has already occurred between two aircraft and collision avoidance becomes paramount.

One system that has been rigorously tested in this manner is the Traffic alert and Collision Avoidance System (TCAS). As part of the TCAS certification process in the 1980s and 1990s, several organizations tested the system across millions of simulated close encounters and evaluated the risk of a near mid-air collision (NMAC, defined as separation less than 500 ft horizontally and 100 ft vertically) [1–4]. This analysis ultimately led to the certification and U.S. mandate for TCAS equipage on larger transport aircraft. More recently, Eurocontrol and the International Civil Aviation Organization (ICAO) performed similar sets of simulation studies for European and worldwide TCAS mandates [5, 6].

The design of a collision avoidance system represents a careful balance to enhance safety while ensuring a low rate of unnecessary maneuvers. This balance is strongly affected by the types of encounter situations to which the system is exposed. It is therefore important that simulated encounters are representative of those that occur in the airspace. Hence, tremendous effort has been made by various institutions since the early 1980s to develop encounter models based on radar data [1, 3, 7–10]. The primary contribution of this report is to introduce a new approach to encounter modeling that is based on a Bayesian statistical framework and which uses recent radar data collected across the U.S. The advantage of applying a Bayesian framework is that it allows us to optimally leverage available radar data to produce a model that is representative of actual operations. The result is a framework from which representative intruder trajectories can be generated and used in fast-time Monte Carlo simulations to provide accurate estimates of collision risk.

A byproduct of the encounter modeling effort was the development of aircraft track and traffic density databases of the National Airspace System. Example plots of traffic density data are provided in this report, but the complete track and density databases are not provided in electronic form due to their size and the complexity of processing specific locations, altitudes, and times.

1.1 ENCOUNTER TYPES

The encounters covered by this model are those involving aircraft in the final stages before a collision. It is assumed that prior safety layers (e.g., airspace structure, Air Traffic Control (ATC) advisories or vectors) have failed to maintain standard separation distances between aircraft. The model is therefore useful in describing the types of situations that need to be addressed by a collision avoidance system, but will not address other separation aspects such as ATC communications or coordination.

Because they are by far the most likely to occur, only pairwise (two-aircraft) encounters are explicitly discussed in this model. If required, the probability of multiple aircraft encounters can be determined from the traffic density database. Random multi-aircraft encounters can be constructed, if desired, by combining several trajectories generated from the model presented here.

There are two fundamental types of encounters. In the first, both aircraft involved are cooperative (i.e., have a transponder) and at least one is in contact with air traffic control. It is then likely that at least one aircraft will receive some notification about the traffic conflict and begin to take action before a collision avoidance system gets involved. We term this type of encounter "correlated" because the trajectories of each aircraft may involve maneuvers that are correlated to some degree due to this prior intervention. The second type of encounter we term "uncorrelated" and involves at least one noncooperative aircraft (i.e., not using a transponder) or two aircraft flying under Visual Flight Rules (VFR) without flight following (i.e., using a transponder Mode A code of 1200). In these encounters, it is unlikely that air traffic control would become involved prior to the close encounter; rather the two aircraft must rely solely on visual acquisition (or some other collision avoidance system) at close range to remain separated. Such encounters tend to be uncorrelated since there is no coordinated intervention prior to the close encounter: the assumption is that the two aircraft blunder into close proximity. A complete evaluation of unmanned systems will require analysis using both correlated and uncorrelated models.

This report focuses on uncorrelated encounter modeling. In this report, we base our uncorrelated model on beacon reports from aircraft squawking 1200, not radar returns from noncooperative traffic. Radar surveillance of noncooperative targets is complicated due to clutter and missed detections, making identification of real tracks difficult. Another modeling effort is underway specifically focused on truly noncooperative aircraft tracks. The lack of a transponder means that only position information, and no identity code or altitude, is available.[1] Hence, it is difficult to infer vertical rates, an important component of an encounter model.

Beacon-equipped aircraft can transmit either a discrete Mode A code or the non-discrete code of 1200. Aircraft using code 1200 tend to be general aviation flying VFR. They generally fly at low altitudes and make significantly more maneuvers than transport aircraft, both horizontally and vertically. Thus to a large degree they resemble noncooperative aircraft. Due to the poor quality of noncooperative data, we use 1200 tracks here as surrogates for primary-only tracks for the construction of this model. Other work underway toward processing true noncooperative tracks will be described in a future document.

Additional analysis (Appendix D) shows that 1200-code tracks are a proper surrogate for certain classes of noncooperative traffic in the National Airspace System, but there are other categories of noncooperative targets for which they are not suitable. For example, most balloons, ultralights, and gliders do not fly like transponder-equipped aircraft squawking 1200. The challenge in developing models for unconventional aircraft such as balloons, ultralights, and gliders is the lack of high-quality data. As these data become available, the same processes outlined in this report will be used to estimate a noncooperative model for unconventional aircraft. Additionally, where such data are not available, a model can be manually constructed based on knowledge of typical flight trajectories and performance limits.

Table 1 shows which encounter model to use depending on the types of aircraft involved in the encounter. There are three types of encounter models shown in this Table: correlated (C), uncorrelated conventional (U), and uncorrelated unconventional (X). In terms of aircraft types, Discrete indicates a cooperative aircraft using a non-1200 Mode A transponder code, and VFR denotes a cooperative aircraft using the 1200 Mode A transponder code. Only the uncorrelated conventional model (bold U) is discussed in this report; the other models are described in other reports.

1.2 RADAR DATA

The radar data used to build the model comes from the 84th Radar Evaluation Squadron (RADES) at Hill AFB, Utah. RADES receives radar data from FAA and Department of Defense sites throughout the United States. They maintain continuous real-time feeds from a network of sensors, including long-range ARSR-4 radars around the perimeter of the United States and short-range ASR-8, ASR-9, and ASR-11 radars in the interior. Radar ranges vary from 60 to 250 NM. Figure 1 shows the approximate coverage by the more than 120 sensors that were used to construct the model. Figure 2 shows the average density of VFR traffic between December 2007 and August 2008, and Appendix H plots flight hour densities at different altitude layers. Appendix G summarizes the volume of data received from each sensor for the construction of the model.

[1] Some military radars have height-finding capability, although the accuracy of the altitude generated is significantly inferior to the transponder Mode C altitude.

TABLE 1

Encounter model types.

C = correlated, U = uncorrelated conventional, X = uncorrelated unconventional

	Aircraft of Interest	
Intruder Aircraft	Discrete	VFR
Discrete	C	C
VFR	C	U
Noncooperative conventional (fixed-wing powered aircraft)	U	U
Noncooperative unconventional (balloon, glider)	X	X

To provide the necessary information to model aircraft trajectories, two weeks of 1200-code aircraft reports were collected from more than 120 sensors, covering the time periods 1–7 December 2007 and 1–7 June 2008. As discussed in Appendix C it was determined that these two weeks of data provided sufficient statistical power and generality to enable a valid and representative model of 1200-code flight characteristics. In addition to this focused two-week data set, all aircraft tracks from the complete nine-month set of radar data were archived and are available for traffic density studies as needed.

Note that there are a number of advantages to the RADES data feed compared to the Enhanced Traffic Management System (ETMS) data often used in airspace analyses. ETMS data include only cooperative aircraft on filed IFR flight plans and provides updates once per minute showing aircraft position after processing by air traffic control automation. In contrast, the RADES data feed streams continuously directly from the radar, including primary-only radar returns as well as cooperative transponder returns (whether on a flight plan or not), providing track updates every 5 or 12 seconds without being affected by automation systems. This ensures that our filters and trackers have the best raw data with which to begin processing.

National Offload Program (NOP) data is another source that could have been used to construct this model. An advantage of NOP data is the inclusion of flight-plan and aircraft-type information. However, NOP data is post-automation, like ETMS, does not include data from Department of Defense sensors, and does not have as comprehensive coverage as the RADES feed.

1.3 PROCESS OVERVIEW

Figure 3 diagrams the steps involved in processing radar data to build the encounter model and generate encounters for simulation. The high-level approach is to model nominal aircraft trajectories using Markov models represented by Bayesian networks (Section 2). The first step in constructing a Markov model involves extracting features, such as turn rate and vertical rate, from the radar data. From these features, sufficient statistics are collected to describe the distribution over maneuvers and other properties of trajectories. Bayesian model selection methods are used to search for the

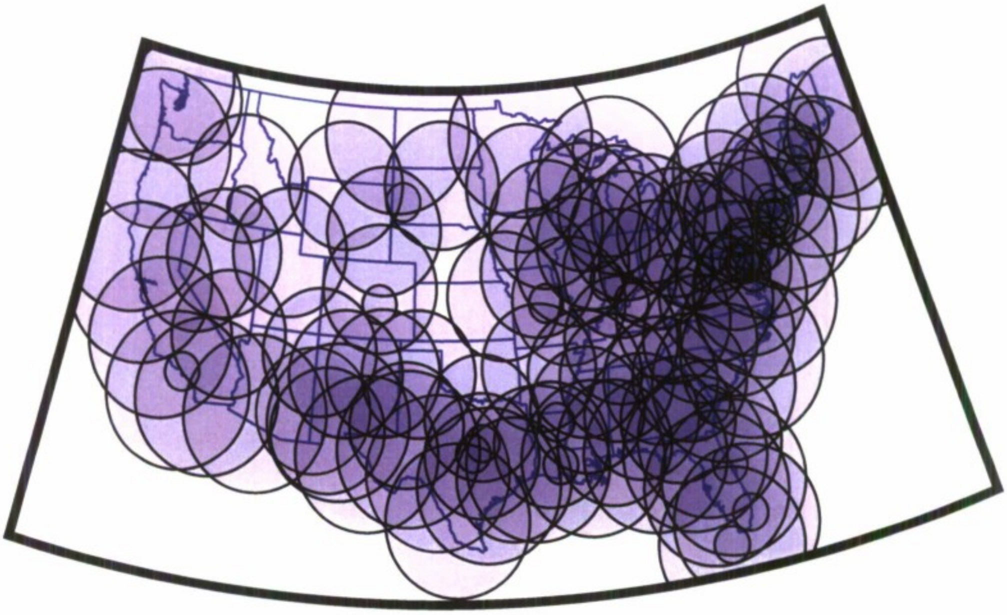


Figure 1. Radar coverage map.

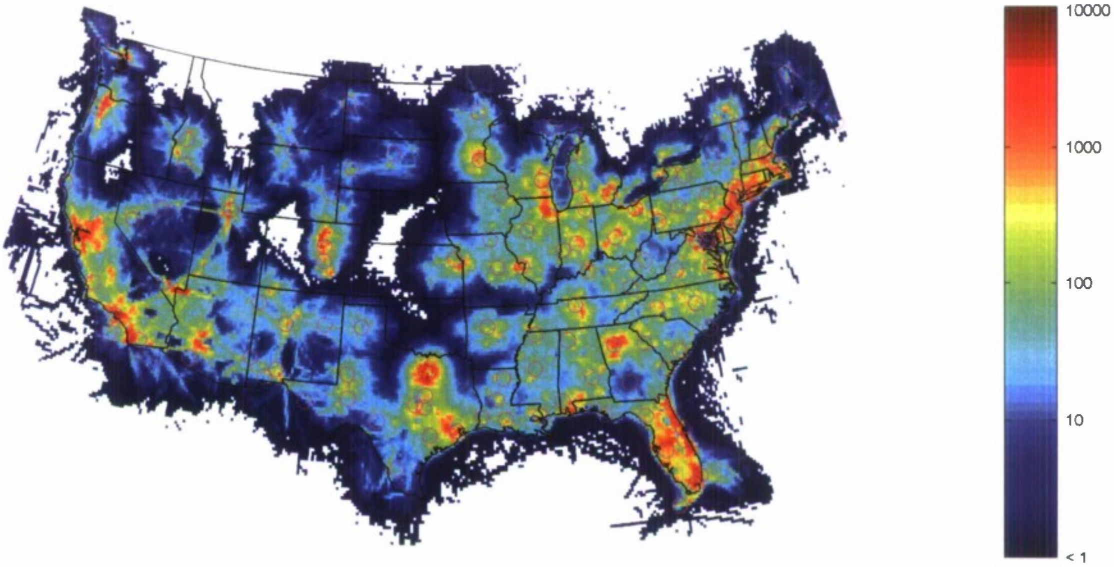


Figure 2. Cumulative VFR (1200-code) flight hours per 1/6 × 1/6 degree cell for the period 1 December 2007 to 31 August 2008. The locations of the sensors are indicated by magenta circles. White indicates no 1200-code reports were found in a cell due to lack of traffic, radar coverage, or terrain masking.

best network structure that represents the observed data (Section 3). The best network structure is then selected and the associated sufficient statistics are obtained to generate new trajectories that are representative of those observed by radar (Section 4). Finally, the trajectories are used in a dynamic simulation to evaluate encounters between aircraft with or without a collision avoidance system (Section 5). Section 6 discusses using the encounter model for large-scale safety analysis and collision risk estimation.

A series of Appendices is also included to provide additional detail and background data. Briefly: Appendix A describes the parameters and data formats used to represent encounters in the model; Appendix B describes how to use the data file and model parameters to generate encounter trajectories; Appendix C describes a comparison of traffic characteristics at varying locations and times during the year; Appendix D discusses the validity of the model toward representing truly noncooperative aircraft; Appendix E describes the process to be followed to validate encounter trajectories generated from the encounter model; Appendix F provides more detail on the radar data processing and track fusion process; Appendix G lists the radars included in the RADES data feed; Appendix H shows traffic density maps for the U.S. at varying altitude levels; Appendix I provides a brief review of Bayesian networks; and finally, Appendix J provides an archive of the different Bayesian networks that were tested as candidates for the final model.

A digital representation of the sufficient statistics and model structure described in this report are available on request from Lincoln Laboratory. An example of using the data in that file to construct a random trajectory is provided in Appendix B. Additionally, a Matlab software package is available to generate random trajectories using the data tables.

Throughout this report we use the standard units used in aviation. In particular, altitudes are in feet, positions are in nautical mile coordinates, vertical rates are in feet per minute, turn rates are in degrees per second, airspeeds are in knots (true airspeed), and accelerations are in knots per second. Time is reported in seconds.

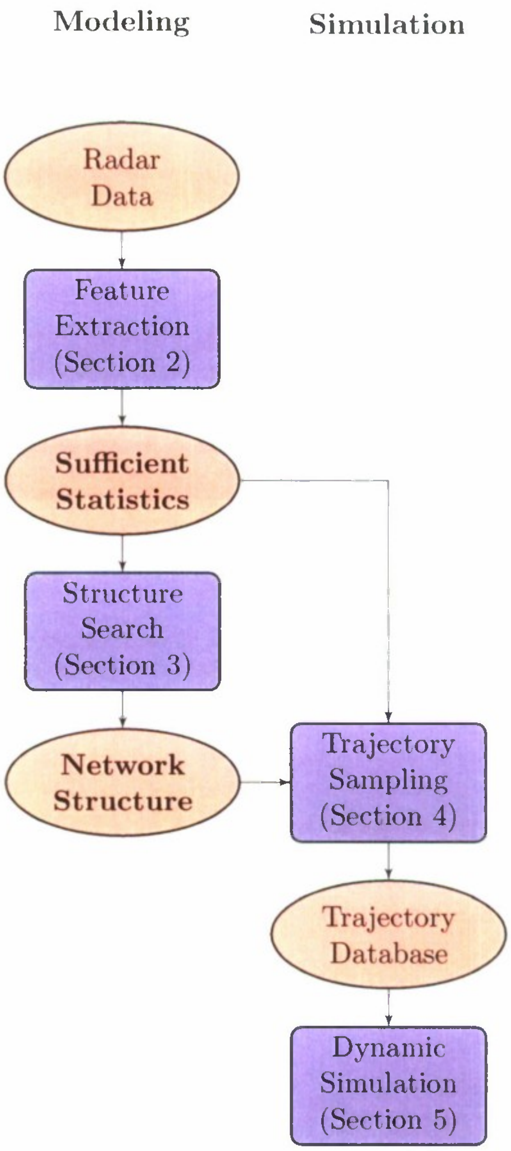

Modeling
Simulation
Radar Data
Feature Extraction (Section 2)
Sufficient Statistics
Structure Search (Section 3)
Network Structure
Trajectory Sampling (Section 4)
Trajectory Database
Dynamic Simulation (Section 5)

Figure 3. Modeling and simulation process overview. The sufficient statistics and network structure (in bold) are the main elements provided as part of this work.

This page intentionally left blank.

2. MODEL

We model nominal flight, i.e., flight without avoidance maneuvering, using a Markov process represented by a dynamic Bayesian network. A Markov process is a stochastic process where the probability distribution over future states is conditionally independent of past states given the present state. In other words, one only needs to know the present state to predict the next state.

The states in the model specify how the position, altitude, and airspeed of each aircraft involved in an encounter change over time. In particular, each state specifies a vertical rate $\dot{h}$, turn rate $\dot{\psi}$, and airspeed acceleration $\dot{v}$. Given an initial airspeed v, position (x, y, h), heading ψ, vertical rate $\dot{h}$, altitude layer L, and airspace class A, one can determine from the model how the aircraft trajectories evolve over the course of the encounter.

One way to represent a Markov model is with an exhaustive state-transition matrix that specifies the probability of transitioning between all pairs of states. Unfortunately, the number of independent parameters (that is, independent probabilities) required to define the matrix grows super-exponentially with the number of variables defining the model. The more independent parameters there are in the model, the more data one needs to properly estimate their values. However, by using dynamic Bayesian networks, we can leverage conditional independence between some variables to greatly reduce the number of parameters. We can learn the structure of the dynamic Bayesian network by maximizing the posterior probability of the network structure given the data.

Appendix I provides the necessary background on Bayesian networks. The remainder of this section defines an encounter in further detail, introduces the variables used to describe an encounter, and presents the modeling structure.

2.1 MODEL VARIABLES

There are six variables in the uncorrelated encounter model:

- **Airspace class** A**:** The airspace class variable was incorporated into the model to account for the variation in how aircraft fly in different airspace classes. This variable may take on one of four values: B, C, D, and O, indicating which class of airspace the aircraft is in. The values B, C, and D correspond to the controlled airspace classes defined by the FAA. The value O represents "other airspace," including airspace Classes E and G. Note that there should be no VFR aircraft in Class A due to the requirement that aircraft in that Class of airspace fly under Instrument Flight Rules.

- **Altitude layer** L**:** Airspace is divided into four altitude layers, in a process similar to prior encounter models developed by Eurocontrol. The first layer spans from 500 to 1200 ft Above Ground Level (AGL) to capture aircraft in the traffic pattern or performing low-level maneuvers. The second layer spans a transition zone from 1200 to 3000 ft AGL, the cruise altitude where the hemispheric rule begins. The third layer spans from 3000 ft AGL to 5000 ft AGL covering a mix of low-altitude enroute and maneuvering aircraft. The fourth layer includes airspace above 5000 ft AGL and would cover most enroute VFR traffic.

- **Airspeed v:** We model true airspeed and allow it to vary during flight.

- **Acceleration $\dot{v}$:** We allow airspeed acceleration to vary every second, providing higher-fidelity motion than prior models.

- **Turn rate $\dot{\psi}$:** Turn rate is permitted to change every second in our model. The prior European and ICAO cooperative models allowed only a single turn during an encounter.

- **Vertical rate $\dot{h}$:** The vertical rate is permitted to change at every second. All prior cooperative models allowed only a single vertical acceleration period during an encounter.

Because many of the variables are closely related due to physical constraints and flight characteristics (e.g., turn rate and vertical rate) is it important to properly represent correlations in the model. Independently sampling from distributions for turn rate and vertical rate would miss these important relationships. The remainder of this section explains how to model joint probability distributions over these variables to ensure proper consideration of correlations. To generate an encounter, we first randomly sample from the joint distribution over the encounter variables to define the encounter geometry and initial conditions. Second, we use a Markov model to determine how the dynamic variables, i.e., turn rate, vertical rate, and airspeed acceleration, evolve during the course of the encounter. There are two corresponding separate probability distributions in the model: an initial distribution to set up an encounter situation, and a transition distribution to describe how the dynamic variables specifying the trajectories of the aircraft evolve over time.

2.2 INITIAL DISTRIBUTION

The aircraft encounter model represents the distribution over the initial values of $\dot{h}$, $\dot{\psi}$, $\dot{v}$, v, L, and A as well as the time-varying history of $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ during the course of an encounter. Probabilistic relationships between these variables are represented using a Bayesian network. An example of such a relationship is the one between turn rate and vertical rate. Without properly capturing this dependency and other important dependencies, unrealistic situations may be generated, e.g., involving aircraft with simultaneously high climb rates and high turn rates. Initial position and altitude is determined through a separate process described in Section 5.

A Bayesian network (e.g., Figure 4a) includes a series of nodes represented by rectangles and directed arcs represented by arrows. Each node corresponds to a particular variable that may take on one of several discrete values with associated probabilities. Certain variables, such as airspace class or altitude layer, are naturally quantized into a few discrete values (e.g., B, C, D, and O for airspace class). Continuous variables, such as vertical rate or turn rate, are described by a series of discrete bins from which a continuous value is later selected. Within each node, then, is a description of the possible values a variable can take and the probability that each value will occur. The directed arcs describe how the probabilities of one variable depend on other variables. Arrows leading into a particular node denote which parent nodes must first be evaluated in order to select the value of the given node. For example, referring to Figure 4a, the probabilities for node L depend on the value of node A; the probabilities for node v depend on the values of nodes A and

L. In the latter case, for instance, there would be a conditional probability table describing the probability of each possible value of v jointly conditioned on the values of A and L: $P(v \mid A, L)$.

Because there are many possible ways variables can be connected in the model, it is necessary to use a quantifiable scoring process to evaluate the quality of each candidate network. For this model, we used a Bayesian scoring process (Appendix I) to evaluate different Bayesian network structures and choose a structure that optimally represents the trajectories we observed. Figure 4a shows the optimal structure for the initial distribution. Appendix J shows other candidate network structures and their scores.

Given a structure, sufficient statistics extracted from data, and a Bayesian prior, we then sample from the Bayesian network to produce initial airspace classes, altitude layers, vertical rates, turn rates, airspeeds, and accelerations that are representative of those found in the data. The nodes and directed arcs used in the structural diagram show the order in which this sampling occurs. For example, based on the structure in Figure 4a, to determine the initial state of the aircraft we first randomly determine an airspace class A. Once the airspace class has been determined, an altitude layer L is selected. The probabilities associated with each altitude layer depend on which airspace class was chosen earlier. Once A and L have been selected, we then randomly select airspeed v, and so on. An example of working through this process is provided in Appendix B. Alternately, we could assign outright an airspace class and altitude layer for a particular study and then randomly select values for the remaining variables.

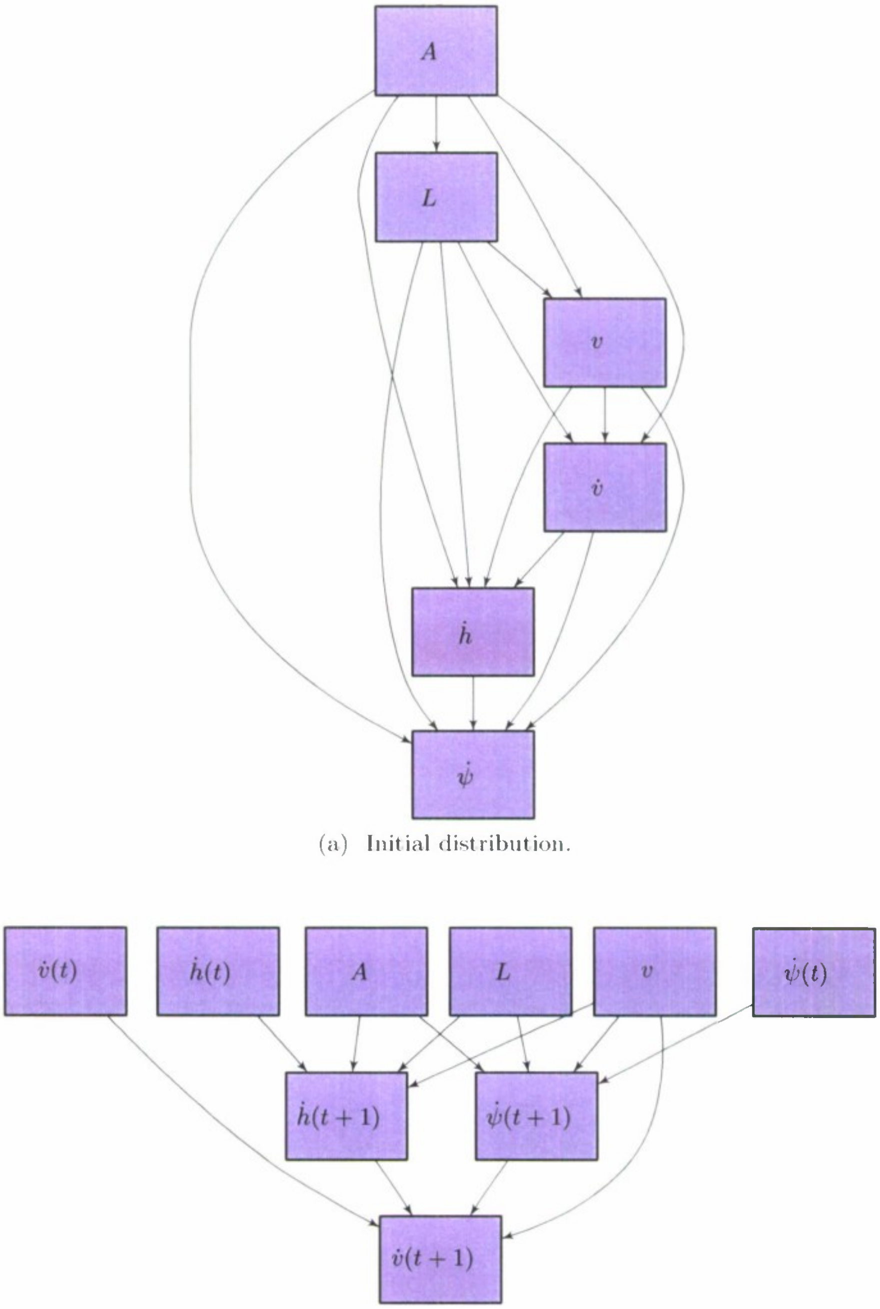
2.3 TRANSITION DISTRIBUTION

We use a separate Bayesian network to model how the variables $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ evolve over time. In this network, we have a first layer that represents the state of the system at the present time step and a second layer that represents the state of the system at the next time step. There are dependencies between layers and within the second layer. Such a two-layer temporal Bayesian network is known as a dynamic Bayesian network [11, 12]. Parameter and structure learning in dynamic Bayesian networks is similar to regular Bayesian networks (Appendix I).

Figure 4b shows the structure used for the transition distribution. Again, we chose the highest-scoring network structure among our candidate network structures (see Appendix J). Given a structure, sufficient statistics extracted from data, and a prior, we can then sample from the Bayesian network to determine the next vertical rate, turn rate, and airspeed acceleration command.

In general, time steps in dynamic Bayesian networks may be of any duration, but for our modeling effort we chose steps of 1 s. Shorter time steps allow for more frequent variations in airspeed, vertical rate, and turn rate, but they require more computation per unit of simulation time. Time steps of 1 s balance maneuver complexity with computation and are also appropriate given typical timescales of dynamic maneuvers.

A complete trajectory is constructed by updating the aircraft state in 1 s intervals. Within each interval, the three derivative variables $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ are treated as target values and held constant. A dynamic model is used to compute and update the aircraft state at each time step based on these piecewise-constant target values. The dynamic model is independent of this encounter model and not discussed in this report, but a process for validating an encounter model against the one used at Lincoln Laboratory is described in Appendix E.



(a) Initial distribution.

(b) Transition distribution

Figure 4. Bayesian networks representing the variable dependency structure for the initial and transition distributions.

This page intentionally left blank.

3. ESTIMATION

At a high level, the modeling process involves taking in a large volume of radar data (spanning two weeks from more than 120 sensors) and carefully filtering and processing that data to extract features of aircraft trajectories. Features include static variables that specify an encounter (such as altitude layer or initial airspeed) and multiple, dynamic variables that describe aircraft motion leading up to and through the closest point of approach (such as turn rate, vertical rate, and airspeed acceleration every second). To aid in data processing, each feature was quantized into several bins and counts were taken of the frequency with which each bin was occupied by radar data. Based on these counts (termed sufficient statistics), probability tables were then constructed so that each feature can be randomly generated such that the overall geometries and dynamics are representative of the actual events observed in the radar data.

Accordingly, the inputs to this process are raw radar reports (range, azimuth, altitude, time) and the outputs are probability tables that specify the likelihood that a given feature will take on a value within a bin corresponding to each table cell. This section describes the processing required to transform radar tracks into sufficient statistics that may be used to model uncorrelated encounters. Figure 5 outlines the multiple-stage feature extraction process.

To build the model, we collected VFR (1200-code) beacon reports between December 1–7, 2007, and between June 1–7, 2008 amounting to 74,000 flight hours after fusing tracks seen by multiple radars. Appendix C describes tests to ensure that the two weeks used to build the uncorrelated model were more than sufficient to build a statistically-valid model for uncorrelated encounters. In addition, nine full months of radar data were archived and used to build distributions of VFR traffic density across the United States. A separate correlated model also developed by Lincoln Laboratory required processing a full nine months of radar data because encounters between two aircraft are relatively rare and because the correlated model has more dependencies between variables.

The raw radar data are first processed using a tracking algorithm developed at Lincoln Laboratory [13]. A fusion algorithm, also developed at Lincoln Laboratory [14], then fuses tracks from multiple sensors to give one global view of all the tracks in U.S. airspace (see Appendix F). We eliminated tracks that had fewer than ten scans. We found that approximately ten scans are required to accurately estimate the various maneuver rates. We also eliminated tracks if any of their associated reports were inside Special Use Airspace whose boundaries are defined in the Digital Aeronautical Flight Information File (DAFIF), 8th Edition, managed by the National Geospatial-Intelligence Agency (NGA).

3.1 OUTLIER REMOVAL

The first step in processing the raw radar tracks is to detect and remove outliers. In the horizontal plane, we remove jumps with ground speeds above 600 kt using the following algorithm. We begin by estimating the speed between each sample point by dividing the distance between samples by the time interval between samples. Samples on either side of segments where the speed is above

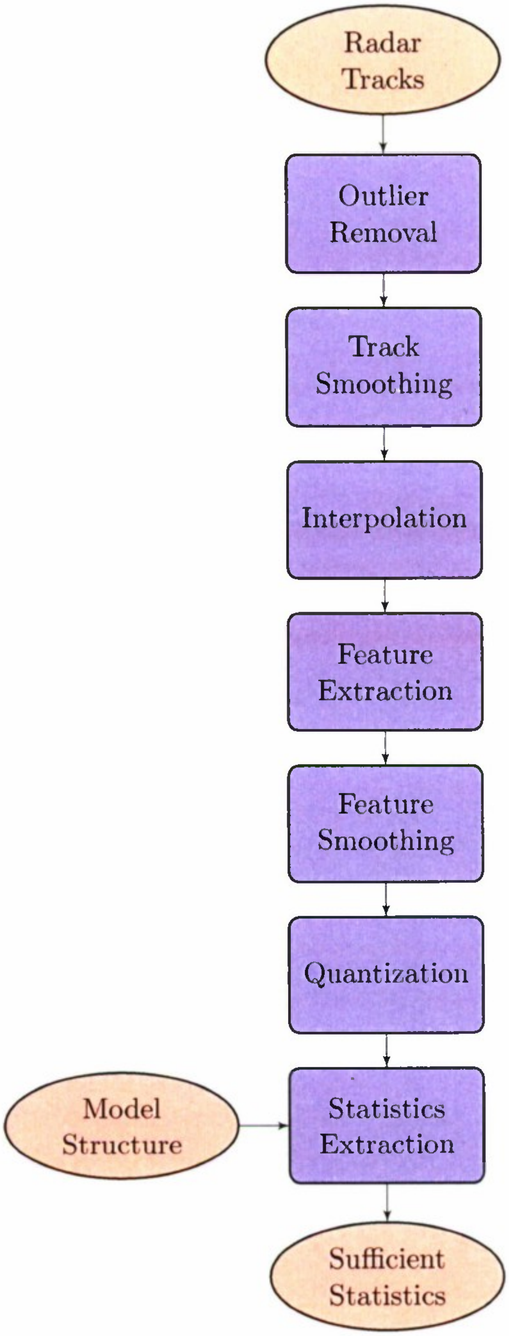


Figure 5. Estimation process flow.

the threshold of 600 kt are stored in a list of candidates for removal. We iterate through the list of candidates and remove the one that minimizes the sum of speeds above the set threshold. The process repeats until there are no longer any segments with speeds above the threshold.

In the vertical plane, we remove missing Mode C altitude reports from consideration. Then, using the same process as used for the horizontal plane, we remove outliers with vertical rates greater than 5000 ft/min or less than −5000 ft/min. We remove altitude reports that come before the first position report or after the last position report to prevent extrapolation. We also remove position reports that come before the first altitude report or after the last altitude report, also to prevent extrapolation. After outlier removal, we discard tracks with fewer than ten valid scans.

3.2 TRACK SMOOTHING

After removing any outliers from a track, we smooth the remaining data points, first horizontally and then vertically. We use the same smoothing scheme for both horizontal and vertical smoothing. We use the following general formula to transform a raw trajectory $(t_1, \boldsymbol{x}_1), \ldots, (t_n, \boldsymbol{x}_n)$ to a smoothed trajectory $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$,

$$\boldsymbol{y}_i = \frac{\sum_j w(t_i, t_j)\boldsymbol{x}_i}{\sum_j w(t_i, t_j)}, \tag{1}$$

where $w(t_i, t_j)$ is a weighting function that monotonically decreases as the difference between t_i and t_j increases. For the weighting function, we use the following definition based on a Gaussian kernel with standard deviation σ,

$$w(t_i, t_j) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t_i - t_j)^2}{2\sigma^2}\right). \tag{2}$$

When smoothing horizontally, we use $\sigma = 5$ s. When smoothing vertically, we use $\sigma = 15$ s. A larger σ is required for vertical smoothing because of 100 ft Mode C quantization. We chose these values for σ after trying different standard deviations on a sampling on horizontal and vertical profiles in our data set; the chosen values preserve the underlying tracks while removing noise.

3.3 INTERPOLATION

The time interval between radar scans in the data is much longer than the 1 s time step of the dynamic Bayesian network. Terminal (short range) radars scan aircraft approximately every 5 seconds, and en route (long range) radars scan aircraft every 10 to 12 seconds. Additionally, it is common for sensors to skip one or more consecutive scans of a target and some scans produce outliers that we remove (Section 3.1). Hence, we need to use interpolation to estimate the parameters in our dynamic Bayesian network. We chose a piecewise-cubic Hermite interpolation scheme that preserves monotonicity and shape [15].

Figure 6 shows the result of outlier detection, smoothing, and interpolation on an example track from the radar data set. Figure 7 shows the result of piecewise-cubic Hermite interpolation on a different example smoothed track.

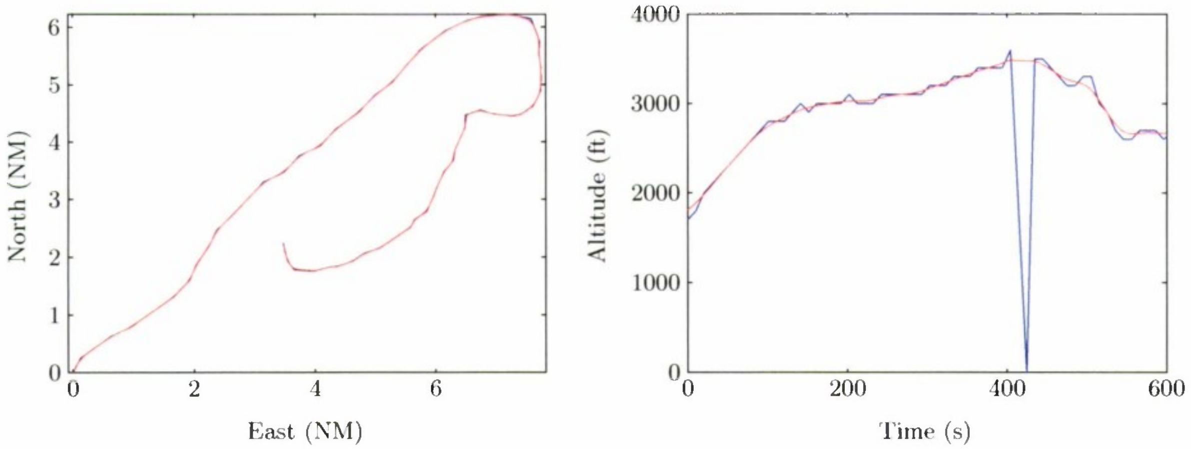


Figure 6. Preprocessing. Blue lines show an example raw track. Red lines show the track after outlier removal, smoothing, and interpolation.

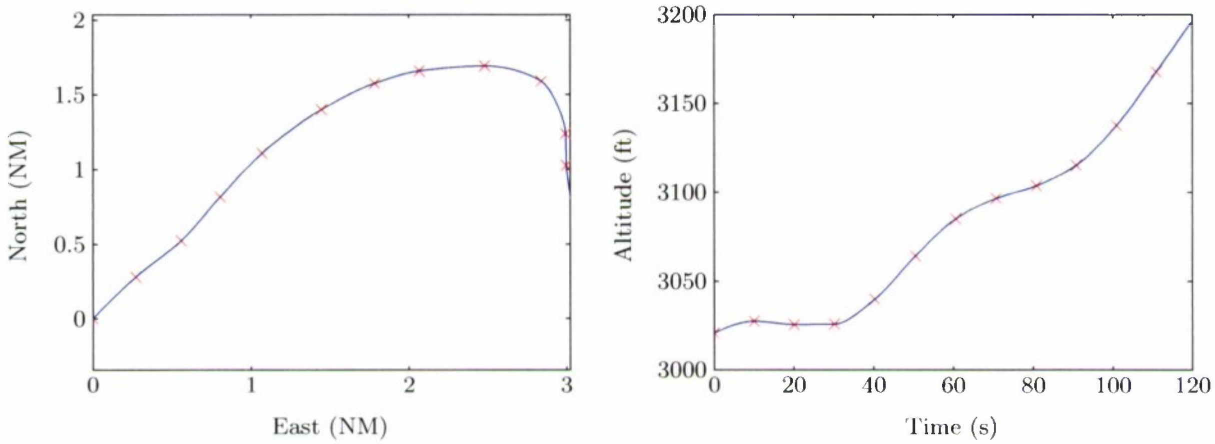


Figure 7. Piecewise-cubic Hermite interpolation on an example smoothed track. Red crosses indicate smoothed data points, and the blue curve shows the interpolation.

3.4 FEATURE EXTRACTION

Feature extraction involves converting an interpolated track into sequences of quantized bins representing airspace classes, altitude layers, airspeeds, vertical rates, turn rates, and accelerations.

- **Airspace class:** We estimate latitude and longitude of radar returns using an algorithm developed at Lincoln [16]. From altitude estimates and latitude and longitude estimates, we determine the class of airspace by searching through the National Airspace System Resources (NASR) database provided by the FAA. Since the altitude estimates are based on Mode C reports of pressure altitude, uncorrected for barometric variation, it is possible that the airspace of some tracks are identified incorrectly. We expect that this limited inaccuracy in airspace class identification due to barometric variation has a negligible impact on our model.

- **Altitude layer:** Altitude above ground level (AGL) determines the altitude layer in the model. We estimate altitude AGL by subtracting an estimate of ground elevation from pressure altitude. Our estimates of ground elevation come from Digital Terrain Elevation Data (DTED) provided by the National Geospatial-Intelligence Agency (NGA). We use DTED Level 0, which has post spacing of 30 arcseconds (approximately 900 meters).

- **Airspeed:** The true airspeed at time t is estimated by

$$v(t) = \sqrt{(x(t+1) - x(t))^2 + (y(t+1) - y(t))^2 + (h(t+1) - h(t))^2}\,.$$

- **Vertical rate:** The vertical rate is estimated from the smoothed and interpolated altitudes estimated from Mode C reports. The vertical rate at time t is given by $\dot{h}(t) = h(t+1) - h(t)$.

- **Turn rate:** We first compute the heading along the interpolated track. The heading at time t is given by $\psi(t)$ and corresponds to the direction from $(x(t), y(t))$ to $(x(t+1), y(t+1))$. To compute the turn rate at time t, we find the acute change in heading between $\psi(t)$ and $\psi(t+1)$. Turns to the right have positive turn rates, and turns to the left have negative turn rates.

- **Acceleration:** To find the acceleration at a particular point, we average the change in airspeed per unit time looking forward one time step and looking back one time step.

We then smooth the extracted features using the same smoothing scheme we used for tracks (Section 3.2). For turn rate, airspeed, and acceleration, we set σ to 10 s, 20 s, and 20 s respectively. We chose these numbers large enough so that noise is removed from the measurements but low enough so that the underlying properties of the maneuvers are not lost. We do not smooth vertical rates in this step because the altitudes are already smoothed (Section 3.2).

In order to be modeled by a discrete Bayesian network, it is necessary to quantize the features. We quantize continuous values by defining a sequence of cut points $c_1, \ldots, c_n$. Values less than c_1 are in the first bin, values greater than c_n are in the $(n+1)$th bin, and values in the half-open interval $[c_{i-1}, c_i)$ are in the ith bin. The cut points we used for quantization are listed in Table 2. For example, referring to Table 2, all airspeed values less than 30 kt are placed into one bin; airspeeds between 30–60 kt are placed in a different bin, etc. The cut points were chosen to capture the variation of the features as shown in the histograms in Figure 8.

Figure 9 shows the result of feature extraction on the same track shown in Figure 6.

3.5 STATISTICS EXTRACTION

With structures for the initial and transition distributions and the quantized features from a set of tracks, we are able to collect the sufficient statistics to estimate the parameters for our model. For the two Bayesian networks, the sufficient statistics are simply the counts N_{ijk} of the various features (see Appendix I).[2] Appendix A describes the sufficient statistics extracted from the data.

[2] The counts are called *sufficient statistics* because they provide a summarization of the data that is sufficient to compute the posterior distribution from the prior. For an introduction to Bayesian statistics, see [17].

TABLE 2

Cut points used for feature quantization.

	Cut Points
A	B, C, D, O
L	$1200, 3000, 5000$
v	$30, 60, 90, 120, 140, 165, 250$
$\dot{v}$	$-1, -0.25, 0.25, 1$
$\dot{h}$	$-1250, -750, -250, 250, 750, 1250$
$\dot{\psi}$	$-6, -4.5, -1.5, 1.5, 4.5, 6$

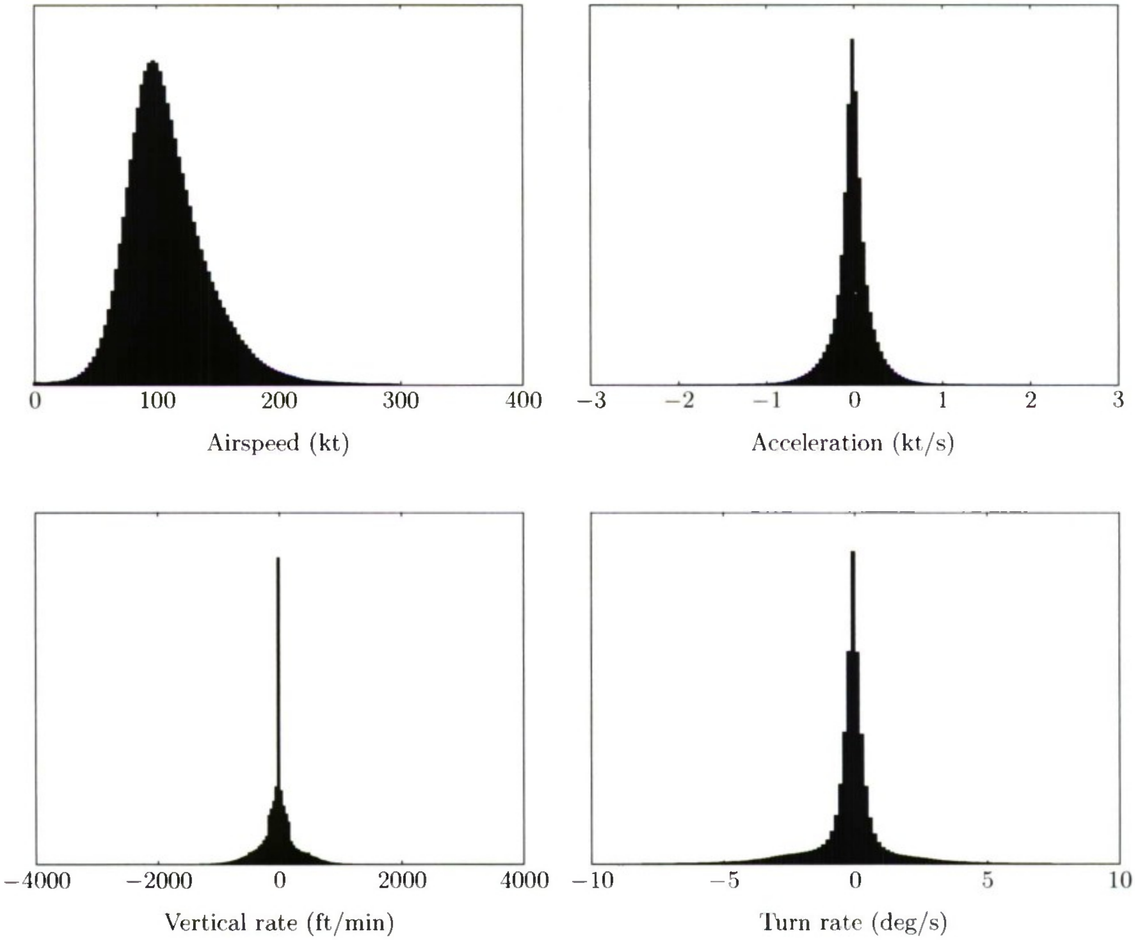


Figure 8. Feature histograms of recorded radar data based on 193 million samples.

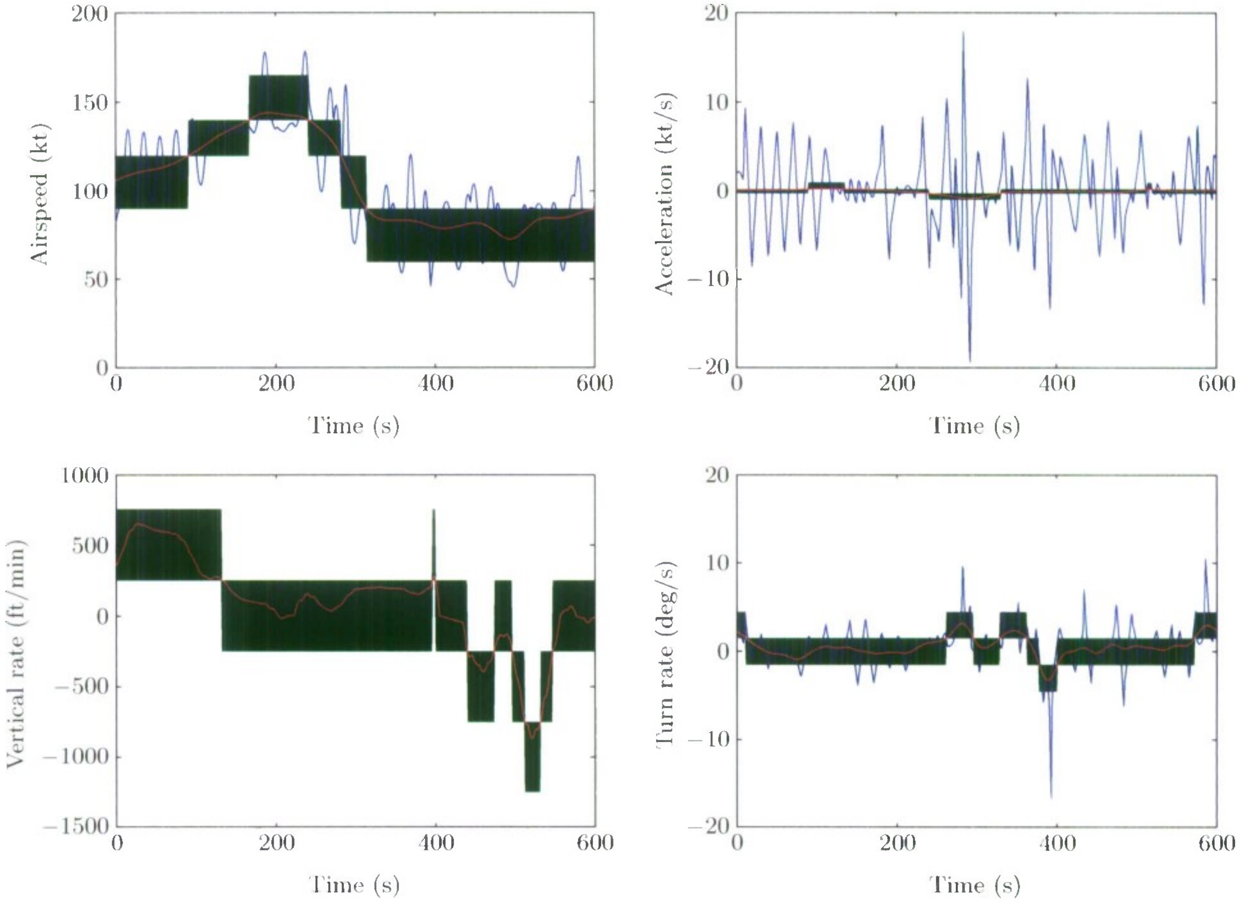


Figure 9. A plot of extracted features over time. Blue lines show features before smoothing, and red lines show features after smoothing. The green blocks show the bins to which the features are assigned.

The counts N_{ijk} are then compiled into a separate table for each variable. For example, the table for altitude layer L (which is itself dependent on airspace class A) is shown in Table 3. Each cell entry in the table represents the counts N_{ijk} for that combination of parent bin A (B, C, D, O) and bin of L ($[500, 1200)$, $[1200, 3000)$, $[3000, 5000)$, $[5000, \infty)$). An electronic file available from Lincoln Laboratory contains all of the data required to construct these tables.

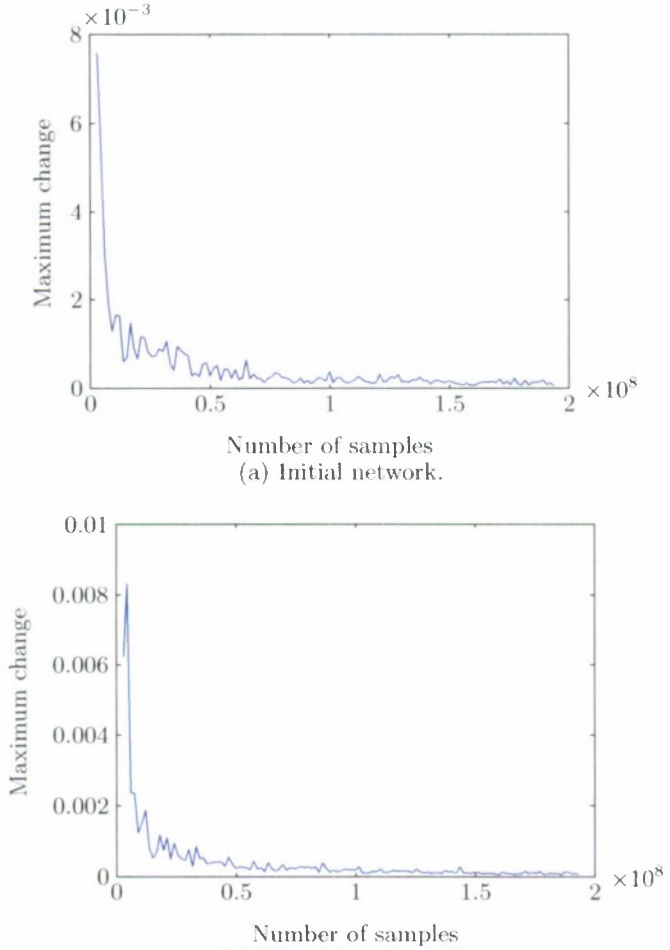
TABLE 3

$N(L \mid A)$

	L			
A	[500, 1200)	[1200, 3000)	[3000, 5000)	[5000, ∞)
B	265497	315803	197208	64204
C	500465	402396	67303	59
D	11209219	4496306	688	0
O	41043613	87491669	31033237	16457213

To ensure a sufficiently large data set was used, Figure 10 illustrates the convergence of the Bayesian network parameters (probabilities) as additional data are added. The horizontal axis represents the number of samples used to estimate the parameters of the Bayesian network, and the vertical axis represents the maximum difference between any elements in the conditional probability tables with the addition of more data. As the figure shows, the change in the probabilities in the conditional probabilities converged to much less than 0.0001 given the amount of radar data collected over the span of two weeks.

To summarize, this section describes the process used to construct a model of uncorrelated aircraft trajectories based on radar data. Raw radar data were processed through multiple filtering and tracking stages and then digested into a series of probability tables organized within Bayesian networks. Each table cell represents the probability of a particular feature taking on a value within a certain quantized bin. The next section describes how to use these tables to generate random, but statistically representative, trajectories through sampling. An example of producing encounter trajectories using the model is given in Appendix B.



(a) Initial network.

(b) Transition network

Figure 10. Convergence curve for Bayesian network conditional probabilities.

This page intentionally left blank.

4. SAMPLING

Once the data have been processed as described in the previous section, one can use the model structure and sufficient statistics to produce new trajectories that are representative of the ones observed by radar. The first step involves sampling from the discrete Bayesian network tables representing the initial and transition distributions. This provides a series of bins that represent coarse values to be used for each feature. The second step involves converting the coarse, discrete samples into fine, continuous samples by sampling within bins. Figure 11 illustrates this process.

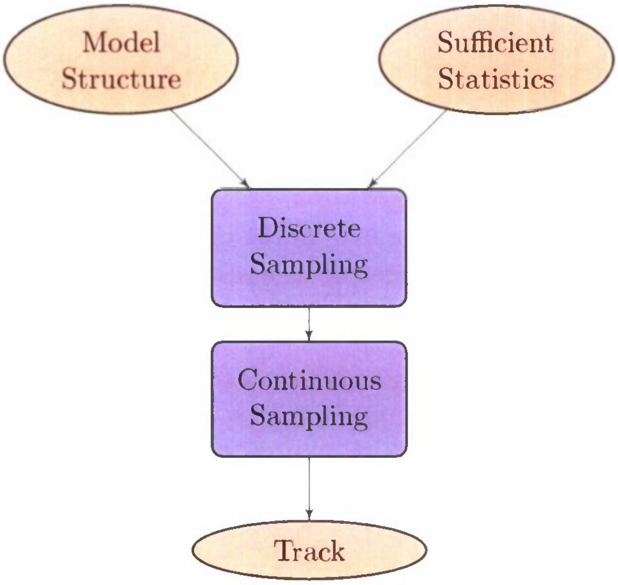


Figure 11. Sampling process flow.

4.1 DISCRETE SAMPLING

We begin by sampling from the Bayesian network representing the initial state distribution. Referring back to Figure 4a, first one would randomly sample from the table describing $N(A)$ to determine the bin to use for airspace class A. Given that bin for airspace class, one would next sample from the table $N(L \mid A)$ to determine the bin for altitude layer L and so on. See Appendix B for a more detailed description of this process.

Formally, consider sampling from a table describing the ith variable X_i (e.g., where X_i = altitude layer). As was shown in Table 3, each table contains a series of bins k (e.g., $[500, 1200)$, $[1200, 3000)$, $[3000, 5000)$, $[5000, \infty)$) whose likelihood depends on each parent bin j (e.g., B, C, D, O).

The probability assigned to bin k is then given by

$$\frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i}(\alpha_{ijk'} + N_{ijk'})}, \tag{3}$$

where

- j is the instantiation of the parents of X_i in the Bayesian network,
- N_{ijk} is the number of times X_i was equal to k when its parents were instantiated to j in the data,
- α_{ijk} is a Dirichlet prior parameter, and
- r_i is the number of ways to instantiate the parents of X_i.

We use $\alpha_{ijk} = 1$, which corresponds to the prior assumption that all combinations of relative frequencies for k are equally probable. Sampling from the posterior distribution with $\alpha_{ijk} = 1$ is equivalent to adding 1 to all the counts in the tables in Appendix A and sampling according to the resulting relative frequencies. This ensures that there are no transitions with zero probability in the Markov model.

For instance, referring to Table 3, if airspace class A had previously been selected to be D, then the probability of selecting altitude layer $L = [500, 1200)$ would be $P(L = [500, 1200)) = (1 + 11209219)/(1 + 11209219 + 1 + 4496306 + 1 + 688 + 1 + 0) = 0.7137$.

Once we have the initial state of the trajectory, we can sample from the dynamic Bayesian network representing how the state changes. We fix $\dot{h}(t)$, $\dot{\psi}(t)$, and $\dot{v}(t)$, and then use the standard Bayesian network sampling scheme to determine $\dot{h}(t+1)$, $\dot{\psi}(t+1)$, and $\dot{v}(t+1)$. The process may be repeated for as long as we wish to run the trajectory.

Our sample from the Bayesian network tells us into which bins airspeed, vertical rate, turn rate, and acceleration fall. We then sample within the bins as discussed in Section 4.2.

4.2 CONTINUOUS SAMPLING

To produce a continuous sample given a coarse, discrete sample from the initial distribution, we simply sample uniformly within the bins. For example, if we determine that the initial airspeed is within the bin $[60, 90)$, we simply sample from the uniform distribution over the half-open interval $[60, 90)$. Because the first and last bins associated with each interval are unbounded, it is necessary to impose some bounds. Table 4 shows the quantization boundaries based on the limits observed in the radar data.

With regard to the transition network, instead of sampling at every time step within bins for turn rate, vertical rate, and airspeed acceleration, we only produce a new sample with a fixed probability per time step that has been estimated from the radar data. This prevents numerous minor changes and allows relatively steady-state flight conditions to occur at the same frequency

TABLE 4

Sampling boundaries.

	Boundaries
h	$500, 1200, 3000, 5000, 12500$
v	$0, 30, 60, 90, 120, 140, 165, 250, 300$
$\dot{v}$	$-2, -1, -0.25, 0.25, 1, 2$
$\dot{h}$	$-2000, -1250, -750, -250, 250, 750, 1250, 2000$
$\dot{\psi}$	$-8, -6, -4.5, -1.5, 1.5, 4.5, 6, 8$

as observed in the radar data. The within-bin transition rates we used are 0.0200, 0.0395, and 0.0875 changes per second for acceleration, vertical rate, and turn rate respectively. We estimated these rates from the data by introducing 3 smaller bins within each bin and computing the relative frequency that tracks stay within a single smaller bin (versus moving to another small bin within the same coarse bin). A similar strategy was used by Eurocontrol for their cooperative encounter model.

When producing continuous samples from bins that include zero in their range, we simply produce zero instead of sampling uniformly in order to prevent very small vertical rates, turn rates, and acceleration.

Figure 12 plots an example of vertical rates, turn rates, accelerations, and airspeeds generated by sampling from the Bayesian networks. First, coarse bins are selected randomly and shown as green bars. Next, fine values within each bin are selected and shown in blue. Note that vertical rate, for example, is held to precisely zero when the bin spans zero, and otherwise the vertical rate may be reselected several times within non-zero bins at the mean rate described above. Figure 13 shows the resulting track produced by the sampled features shown in Figure 12. Translation of features into tracks involves running a discrete-time simulation, described in the next section.

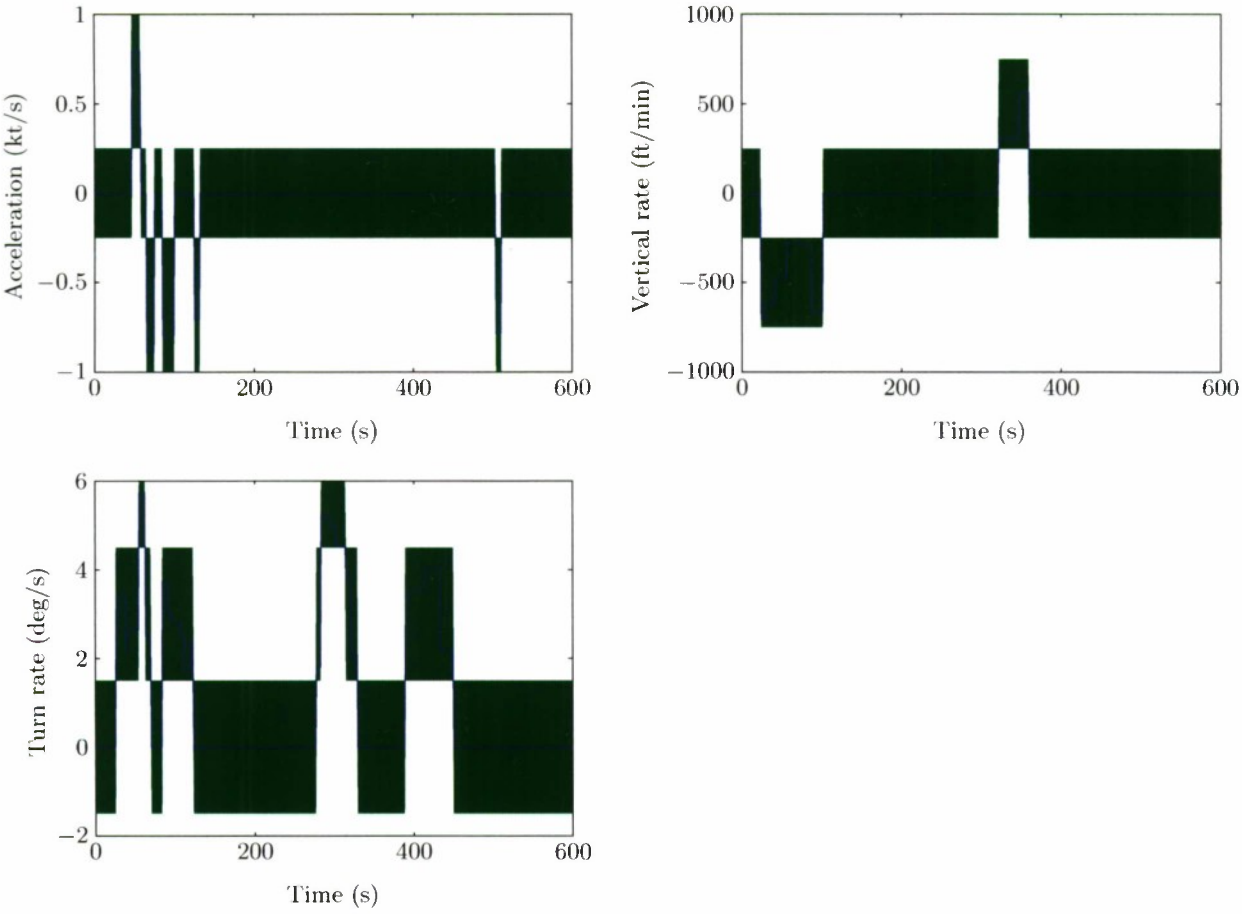


Figure 12. A plot of sampled features over time. Green blocks show the coarse discrete bins and blue lines show the resulting fine samples within those bins.

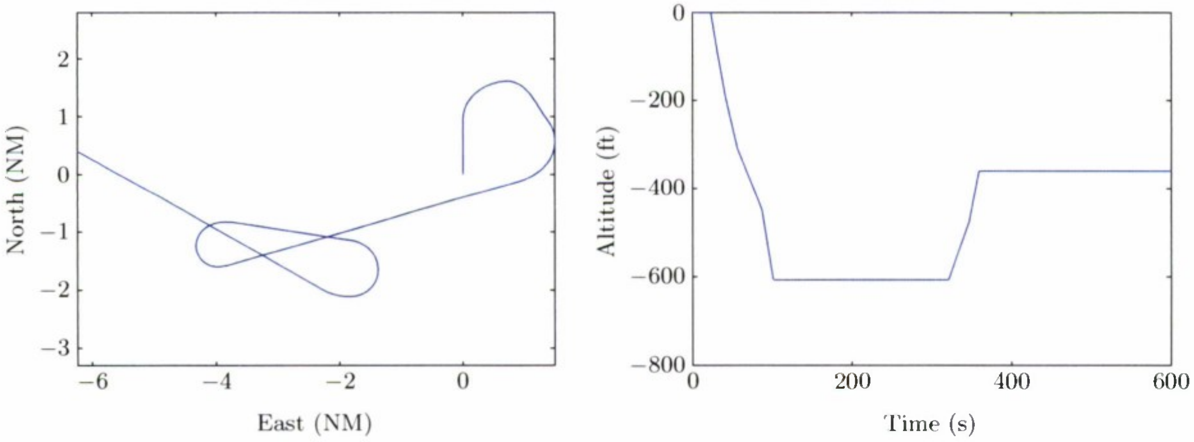


Figure 13. A track generated by sampling from the initial and transition distributions. Positions and altitudes are shown relative to the initial position and altitude.

5. SIMULATION

Earlier sections explained how to build a dynamic probabilistic model of aircraft and how to sample from the model to produce nominal trajectories that are representative of what we observed in the radar data. The same modeling techniques discussed earlier may be used to build a model of the manned or unmanned aircraft with the collision avoidance system to be evaluated. This section explains how to combine a model of the aircraft with the collision avoidance system to be evaluated, which we call AC1, with a model of an intruder, which we call AC2, into a simulated encounter.

The trajectory for AC1 may be specified by the analyst (e.g., to focus on a particular phase of flight), based on actual flight paths from mission planning or radar data, or randomly generated using a statistical model representative of that aircraft's typical flight profiles. In a study of conventional VFR-on-VFR encounters, for example, trajectories for both AC1 and AC2 could be generated using the uncorrelated encounter model described here.

We avoid simulating encounters that are extremely unlikely to result in NMACs by focusing computational effort on encounters that occur in an encounter cylinder centered on AC1. AC2 is initialized at a random location on the surface of the encounter cylinder and the dynamic models are used to update the states of AC1 and AC2 over time. If the collision avoidance system on AC1 commands an avoidance maneuver, it overrides the nominal rates suggested by the dynamic model. If AC2 enters an NMAC cylinder or exits the encounter cylinder, the encounter run is terminated. The NMAC cylinder has radius r_{nmac} and height $2h_{\text{nmac}}$. We use $r_{\text{nmac}} = 500\,\text{ft}$ and $h_{\text{nmac}} = 100\,\text{ft}$.

5.1 ENCOUNTER CYLINDER DIMENSIONS

The encounter cylinder has radius r_{enc} and height $2h_{\text{enc}}$. The appropriate dimensions of the encounter cylinder depend on the aircraft dynamics and collision avoidance system. If the encounter cylinder is too small, the collision avoidance system will not have enough opportunity to be fully exercised before a collision. If the encounter cylinder is too large, then computation is wasted.

An upper bound for r_{enc} is the amount of time required by the collision avoidance system to detect, track, and avoid a target multiplied by the sum of the maximal airspeeds of AC1 and AC2. An upper bound for h_{enc} is the amount of time required by the collision avoidance system to detect, track, and avoid a target multiplied by the sum of the maximal vertical rate magnitudes of AC1 and AC2.

5.2 INITIAL CONDITIONS

We use rejection sampling to generate the initial conditions of an encounter. Rejection sampling involves proposing a series of candidate samples from a random distribution until choosing one that meets a set of criteria (summarized in Figure 14). The process we use for generating initial conditions for encounters is as follows:

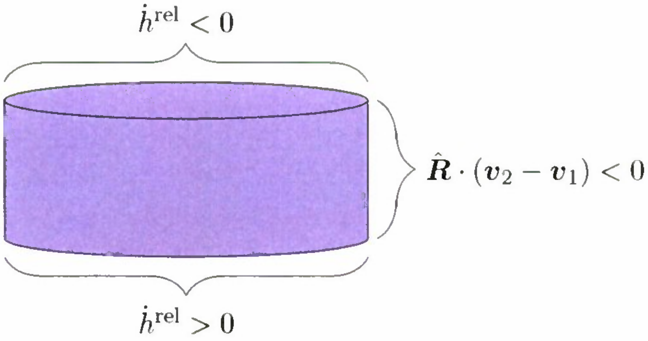

$\dot{h}^{\text{rel}} < 0$
$\hat{\boldsymbol{R}} \cdot (\boldsymbol{v}_2 - \boldsymbol{v}_1) < 0$
$\dot{h}^{\text{rel}} > 0$

Figure 14. Depending on where on the encounter cylinder the intruder is initialized, different criteria must be met in order to accept a given trajectory for further simulation.

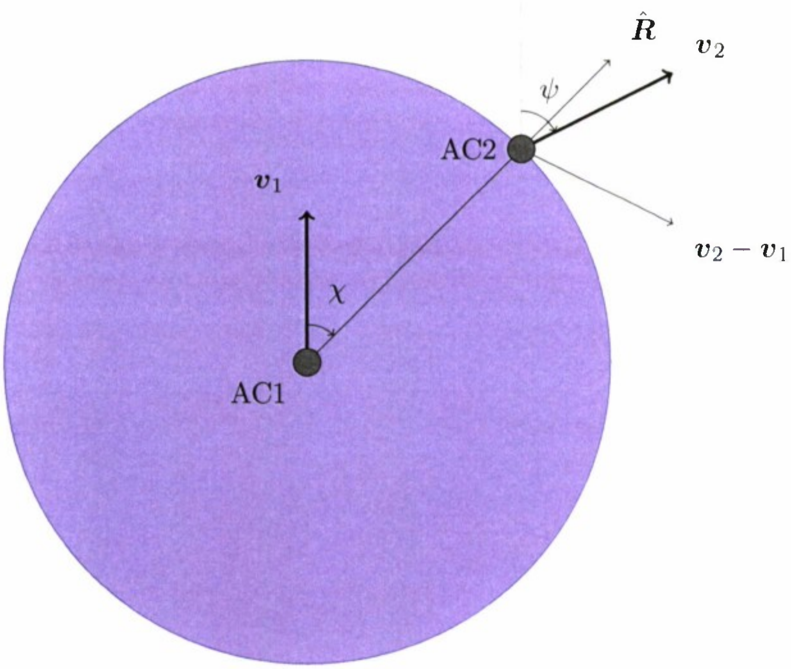


Figure 15. Variables relevant to rejection sampling in the horizontal plane.

1. Generate airspeeds, vertical rates, turn rates, and accelerations for AC1 and AC2 according to their models such that they belong to the same airspace class and altitude layer. Forcing this constraint can be done using rejection sampling. Simply generate AC1 and AC2 independently and reject both if they have a different airspace class or altitude layer.

2. Initialize the position of AC2 on the surface of the encounter cylinder centered on AC1. AC2 may be initialized on the top, bottom, or side surfaces of the encounter cylinder. The probability of being located on the top, bottom, or side is proportional to the surface area of each. Once top, bottom, or side has been selected, AC2 is randomly positioned using a two-dimensional uniform distribution across that surface. The bearing of AC2 relative to AC1 is denoted χ.

3. The heading of AC1 is set to 0. The heading of AC2, denoted ψ, is randomly selected from a uniform distribution over $[0, 2\pi)$.

4. If AC2 was initialized on the top of the encounter cylinder, accept the sample if the vertical rate of $AC2$ relative to $AC1$, denoted $\dot{h}^{\text{rel}}$, is negative. This ensures that AC2 is penetrating the encounter cylinder for the first time.

5. If AC2 was initialized on the bottom of the encounter cylinder, accept the sample if the vertical rate of $AC2$ relative to $AC1$, denoted $\dot{h}^{\text{rel}}$, is positive. This ensures that AC2 is penetrating the encounter cylinder for the first time.

6. If AC2 was initialized on the side of the encounter cylinder, accept the sample if $\hat{\boldsymbol{R}} \cdot (\boldsymbol{v}_2 - \boldsymbol{v}_1)$ is negative, where $\hat{\boldsymbol{R}} = (\sin\chi, \cos\chi)$ is the unit vector from AC1 to AC2. The vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the ground velocities of AC1 and AC2 respectively. When $\hat{\boldsymbol{R}} \cdot (\boldsymbol{v}_2 - \boldsymbol{v}_1)$ is negative, the relative velocity of AC2 is into the encounter cylinder, and therefore should be accepted. See Figure 15.

The process is repeated until a candidate initialization is accepted.

A byproduct of rejection sampling is that the intruder bearing distribution is nonuniform. As one would expect, more encounters occur head on than from the side or rear. Figure 16 shows bearing distributions for two conventional VFR aircraft.

5.3 TRAJECTORY CONSTRUCTION

Once the initial conditions are selected, the dynamic models of AC1 and AC2 are used to update their trajectories over time.

Given the initial state of an aircraft (including at a minimum: position, altitude, heading, climb rate, turn rate, and velocity) and its control variables ($\dot{h}$, $\dot{\psi}$, and $\dot{v}$) the aircraft state is updated in ≤ 1 s time steps using a dynamic model. Due to the wide variety of possible dynamic model implementations [18], details for computing this state update are not provided here. A basic approach would be to apply simple point-mass kinematics to update the aircraft state without considering what the aircraft may actually be doing in terms of bank angle, pitch rate, etc. More

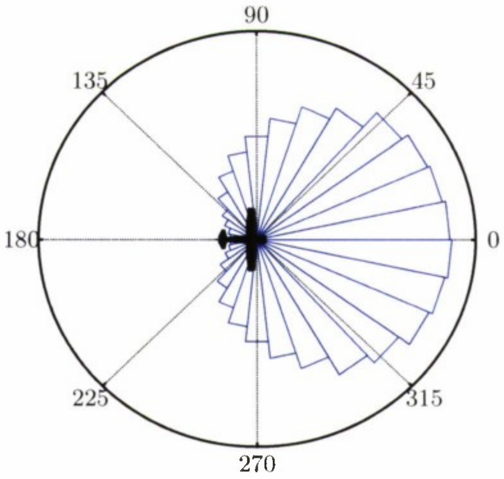


Figure 16. Initial bearing distributions produced by rejection sampling for two randomly-generated VFR aircraft.

complex implementations could include 6 degree-of-freedom dynamic models in which the control variables are treated as target states provided to an autoflight control system which then applies the necessary control deflections to attain those targets. Lincoln Laboratory's Collision Avoidance System Safety Assessment Tool (CASSATT) typically uses a 4 degree-of-freedom model to update aircraft state by applying the necessary airspeed acceleration, roll rate, and pitch rate to achieve the target values for $\dot{h}$, $\dot{\psi}$, and $\dot{v}$ assuming curvilinear motion with a zero-sideslip constraint. The zero-sideslip constraint can be relaxed, for instance, if it is necessary to model transient dynamics with higher fidelity. Appendix E explains how to validate trajectories produced by other simulation software against trajectories produced by CASSATT.

A simulation run terminates when the intruder either exits the encounter cylinder or penetrates the NMAC cylinder. After running many simulations, $P(\text{nmac} \mid \text{enc})$ is estimated by dividing the number of runs that resulted in an NMAC by the total number of runs.

5.4 MULTIPLE ENCOUNTERS

The simulation currently only handles pairwise encounters. The probability of an intruder penetrating the encounter cylinder while another intruder is within the encounter cylinder is likely to be very small. One may compute this probability using

$$\int_0^\infty p(t)[1 - e^{\lambda_{\text{enc}} t}]\mathrm{d}t = 1 - \int_0^\infty p(t)e^{\lambda_{\text{enc}} t}\mathrm{d}t\,, \tag{4}$$

where $p(t)$ is the distribution over the amount of time intruders spend in the encounter cylinder and λ_{enc} is the rate at which new intruders penetrate the encounter cylinder. In the above equation, $1 - e^{\lambda_{\text{enc}} t}$ comes from the cumulative density function for an exponential distribution. The possibility of simultaneous multiple intruders needs to be examined and will be an area of future work.

6. SAFETY EVALUATION

This section explains how to estimate the NMAC rate, denoted λ_{nmac}, based on a large number of simulations. We begin by observing that

$$\lambda_{\mathrm{nmac}} = P(\mathrm{nmac} \mid \mathrm{enc})\lambda_{\mathrm{enc}} ,$$

where $P(\mathrm{nmac} \mid \mathrm{enc})$ is the probability that an aircraft that enters the encounter cylinder penetrates the NMAC cylinder before exiting the encounter cylinder and λ_{enc} is the rate at which aircraft penetrate the encounter cylinder. The mean time between NMACs is simply $\lambda_{\mathrm{nmac}}^{-1}$.

This section makes the following assumptions:

1. the density of air traffic outside the encounter cylinder is uniform in the local region being studied, and

2. the trajectories of aircraft outside of the encounter cylinder are independent of the trajectories of aircraft within the encounter cylinder,

From these two assumptions, we explain how to compute $P(\mathrm{nmac} \mid \mathrm{enc})$ and λ_{enc}. Note that estimating traffic density requires a more focused assessment of a particular region and time period, beyond the scope of this report. However, sufficient radar data have been archived at Lincoln Laboratory for the U.S. to allow such an analysis at most regions (with the exception of potential terrain masking of radar reports in certain areas).

6.1 ESTIMATING NMAC PROBABILITY

Section 5.2 explained how to construct an encounter from two independent trajectories sampled from the distribution represented by Bayesian networks. By generating a large collection of encounters and determining which encounters lead to NMACs, we can estimate $P(\mathrm{nmac} \mid \mathrm{enc})$. Unfortunately, we cannot simply divide the number of sampled encounters that lead to NMACs by the total number of sampled encounters to estimate $P(\mathrm{nmac} \mid \mathrm{enc})$ due to the fact that our sampling scheme does not produce encounters from the same distribution that would occur in the airspace. In particular, we generate encounters with aircraft velocities distributed identically to the aircraft population at large, despite the fact that in reality the distribution of aircraft velocities given that an encounter is occurring favors high-speed aircraft. Although we sample from a distribution that is different from the true distribution when constructing encounters, we can still use the samples to estimate $P(\mathrm{nmac} \mid \mathrm{enc})$ so long as we weight their results properly using an approach known as importance sampling [19]. We will begin by stating the weighting scheme and then prove that it is correct.

Section 4 explained how to generate the trajectories for AC1 and AC2, which we call $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, by sampling from the Bayesian networks with the requirement that both aircraft come from the same airspace class and altitude layer. Section 5.2 explained how to randomly select the position of AC2 relative to AC1, which we call $\boldsymbol{x}_r$, and the heading of AC2 relative to AC1, which we call ψ. Importance sampling allows us to make the following approximation based on N samples

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N}\sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc})\frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}}\,.$$

The weight $V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})/\bar{V}$ corrects for the fact that our sampling distribution does not match the true distribution of encounter situations. The function $V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})$ is the average volume the encounter cylinder sweeps out per unit time when AC1 follows $\boldsymbol{z}_1^{(i)}$ and the airspace consists exclusively of aircraft following $\boldsymbol{z}_2^{(i)}$. In particular,

$$V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) = 4r_{\text{enc}}h_{\text{enc}}\underbrace{\frac{1}{\pi}\int_0^\pi \sqrt{(v_1^g + v_2^g\cos\psi)^2 + (v_2^g\sin\psi)^2}\,\mathrm{d}\psi}_{\substack{\text{average}\\\text{relative}\\\text{horizontal}\\\text{speed}}} + \pi r_{\text{enc}}^2\underbrace{\left|\dot{h}_1 - \dot{h}_2\right|}_{\substack{\text{average}\\\text{relative}\\\text{vertical}\\\text{speed}}},$$

where ψ is the relative heading of AC2, v_1^g and v_2^g are the initial ground speeds, and $\dot{h}_1$ and $\dot{h}_2$ are the initial vertical rates of $\boldsymbol{z}_1^{(i)}$ and $\boldsymbol{z}_2^{(i)}$. The constant $\bar{V}$ is the average volume the encounter cylinder sweeps out per unit time

$$\bar{V} = \iint p(\boldsymbol{z}_1)p(\boldsymbol{z}_2 \mid \boldsymbol{z}_1)V(\boldsymbol{z}_1, \boldsymbol{z}_2)\,\mathrm{d}\boldsymbol{z}_1\mathrm{d}\boldsymbol{z}_2\,.$$

Note that the distribution over $\boldsymbol{z}_2$ is conditional on $\boldsymbol{z}_1$ due to the constraint that AC1 and AC2 must belong to the same airspace class and altitude layer. The constant $\bar{V}$ can be estimated using N samples:

$$\bar{V} \approx \frac{1}{N}\sum_i V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})\,. \tag{5}$$

Now that we have defined our weighting scheme, we will now prove that it is correct. From the laws of probability,

$$\begin{aligned} P(\text{nmac} \mid \text{enc}) &= \iint P(\text{nmac} \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \text{enc})p(\boldsymbol{z}_1, \boldsymbol{z}_2 \mid \text{enc})\,\mathrm{d}\boldsymbol{z}_1\mathrm{d}\boldsymbol{z}_2 \\ &= \iiiint P(\text{nmac} \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \psi, \boldsymbol{x}_r, \text{enc})p(\psi, \boldsymbol{x}_r \mid \boldsymbol{z}_1, \boldsymbol{z}_2, \text{enc})p(\boldsymbol{z}_1, \boldsymbol{z}_2 \mid \text{enc})\,\mathrm{d}\boldsymbol{z}_1\mathrm{d}\boldsymbol{z}_2\mathrm{d}\psi\mathrm{d}\boldsymbol{x}_r\,. \end{aligned}$$

We may approximate $P(\text{nmac} \mid \text{enc})$ using Monte Carlo sampling. Since it is difficult to sample from $p(\boldsymbol{z}_1, \boldsymbol{z}_2 \mid \text{enc})$ directly, we sample $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ from the distribution represented by our Bayesian network subject to the constraint that both aircraft come from the same airspace class and altitude layer, and weight the samples appropriately:

$$P(\text{nmac} \mid \text{enc}) \approx \frac{1}{N}\sum_i P(\text{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \text{enc})\frac{p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \text{enc})}{p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})}\,.$$

We know that

$$\begin{aligned} p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \mathrm{enc}) &= \frac{p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})}{\lambda_{\mathrm{enc}}} \lambda_{\mathrm{enc} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}} \\ &\propto p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) \lambda_{\mathrm{enc} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}} \\ &\propto p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) . \end{aligned}$$

We may normalize to obtain

$$\begin{aligned} p(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)} \mid \mathrm{enc}) &= p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \iint p(\boldsymbol{z}_1)p(\boldsymbol{z}_2 \mid \boldsymbol{z}_1) V(\boldsymbol{z}_1, \boldsymbol{z}_2) \, \mathrm{d}\boldsymbol{z}_1 \mathrm{d}\boldsymbol{z}_2 \\ &= p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \bar{V} . \end{aligned}$$

We may substitute and simplify to obtain

$$\begin{aligned} P(\mathrm{nmac} \mid \mathrm{enc}) &\approx \frac{1}{N} \sum_i P(\mathrm{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \mathrm{enc}) \frac{p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)}) V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}) / \bar{V}}{p(\boldsymbol{z}_1^{(i)})p(\boldsymbol{z}_2^{(i)} \mid \boldsymbol{z}_1^{(i)})} \\ &\approx \frac{1}{N} \sum_i P(\mathrm{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \mathrm{enc}) \frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}} , \end{aligned}$$

which corresponds to the weighting scheme we defined.

6.2 ESTIMATING ENCOUNTER RATE

Estimating the encounter rate requires knowing the density of traffic outside the encounter cylinder. This density, ρ, can be expressed in aircraft per NM^3. The rate at which aircraft enter the encounter cylinder is the product of ρ and the average volume of new airspace the encounter cylinder sweeps through per unit time, $\bar{V}$, which was discussed in Section 6.1:

$$\lambda_{\mathrm{enc}} = \rho \bar{V} . \tag{6}$$

Appendix H plots values for ρ for VFR (1200-code) traffic over the continental United States. The highest density area covers Miami-Dade County and has a density of approximately 0.003 aircraft per NM^3 over all altitude layers between 500 ft AGL and FL180. This density is an average over the data collected from 1–7 December and 1–7 June. It is important to note that the density during the day is significantly higher than at night. Hence, if one is interested in estimating collision risk due to VFR aircraft when flying a particular unmanned aircraft during the day, for example, then one must use a value for ρ that is specific to the expected hours of operation so that collision risk is not underestimated.

The $\bar{V}$ in Equation 6 depends upon the size of the encounter cylinder and on the average velocities of aircraft involved in encounters. As one example, an estimate of $\bar{V}$ was obtained empirically by generating 1 million encounters from the uncorrelated encounter model and applying the approximation in Equation 5. If the encounter cylinder has radius 5 NM and height 3000 ft, then $\bar{V}$ was found to be approximately 937 NM^3/hr for trajectories generated from the uncorrelated encounter model. The average VFR traffic encounter rate for Miami-Dade County, for example, is then $0.003 \times 937 = 2.8$ encounters per flight hour. Note that this does not imply that collision avoidance maneuvering would necessarily occur 2.8 times per hour because not all encounters require maneuvering. Most of these encounters dissipate benignly, but it is important to simulate them to ensure the collision avoidance system does not induce problems. Further, because this example density estimate was based on an average over two weeks, night and day, at all altitudes between 500 to 18,000 ft, it is important to reiterate the need to obtain density values for specific operating altitudes, airspaces, and times to get a more accurate estimate of the encounter rate for a particular operational concept. Finally, encounter rates for cooperative traffic and for non-cooperative unconventional aircraft also need to be considered to obtain an overall collision risk estimate.

We may compute λ_{nmac} as follows:

$$\begin{aligned}
\lambda_{\mathrm{nmac}} &= P(\mathrm{nmac} \mid \mathrm{enc})\lambda_{\mathrm{enc}} \\
&= \rho\bar{V}P(\mathrm{nmac} \mid \mathrm{enc}) \\
&\approx \frac{\rho\bar{V}}{N}\sum_i P(\mathrm{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \mathrm{enc})\frac{V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})}{\bar{V}} \\
&= \frac{\rho}{N}\sum_i P(\mathrm{nmac} \mid \boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)}, \psi^{(i)}, \boldsymbol{x}_r^{(i)}, \mathrm{enc})V(\boldsymbol{z}_1^{(i)}, \boldsymbol{z}_2^{(i)})\,.
\end{aligned}$$

7. SUMMARY

This report presents a new approach to encounter modeling that allows for the generation of more realistic encounters than previous models. The approach involves modeling the dynamics of aircraft state based on a Markov model where the probability of the next state depends only upon the current state. One way to represent a Markov model is with an exhaustive state-transition matrix that specifies the probability of transitioning between all pairs of states. Unfortunately, the number of independent parameters required to define the matrix grows super-exponentially with the number of variables defining the model. The more independent parameters there are in the model, the more data one needs to properly estimate their values. However, by using dynamic Bayesian networks, we leverage conditional independence between some variables to greatly reduce the number of parameters. We learn the structure of the dynamic Bayesian network by maximizing the posterior probability of the network structure given the data.

The work presented in this report has focused on a model where the trajectories of the aircraft involved in the encounter are independent of each other prior to intervention by a collision avoidance system, human or automated. This model assumes that aircraft blunder into close proximity without prior intervention. A separate, correlated encounter model has also been developed for aircraft under air traffic control, where intervention may impact the way in which aircraft encounter each other. Such a correlated encounter model is similar to the uncorrelated encounter model described here except that the variables defining the state of the aircraft in the encounter are tied to each other in the dynamic Bayesian network. We are also developing a noncooperative model of unconventional aircraft such as balloons, ultralights, and gliders.

This page intentionally left blank.

APPENDIX A
MODEL PARAMETERS

This appendix describes the sufficient statistics, N_{ijk} (see Appendix 1), used to estimate the conditional probabilities associated with the initial and transition distributions. These sufficient statistics are based on beacon reports associated with aircraft squawking VFR (Mode A code 1200) from 1–7 December 2007 and 1–7 June 2008. This amounts to 74,000 flight hours from across the United States after track fusion. Other parameters that are relevant to generating new encounters from the model are also described in this section.

A text file, available electronically from Lincoln Laboratory, describes the following model parameters:

- **Variable labels:** A quoted, comma-delimited list specifies the variable labels, e.g., `\dot \psi`, as would be used by LaTeX. There are different variable labels for the initial network and the transition network. The ordering of the variables in this list determines the ordering of the variables in the other tables. Note that the ordering of the variable labels does not necessarily correspond to the order in which they are sampled; a topological sort may be necessary before sampling.
- **Graphical structure:** A binary matrix is used to represent graphical structure. A `1` in the ith row and jth column means that there is a directed edge from the ith variable to the jth variable in the Bayesian network, the ordering of the variables are as defined in the variable labels section of the file. The text file specifies two graphical structures: one for the initial network and the other for the transition network. The element in the ith row and jth column is represented by $G(i, j)$.
- **Variable instantiations:** For each network, a list of integers specifies the number of instantiations that exist for each variable.
- **Sufficient statistics:** For each network, a list of integers specifies the sufficient statistics. We explain how to interpret the array of integers below.
- **Boundaries:** The boundaries of the variable bins are specified by a row of numbers. The variables A and L are not quantized because they are already discrete, and so boundaries do not exist. A `*` is used for these variables.
- **Resampling rates:** A list of numbers specify the resampling rates (Section 4.2).

The list of numbers describing the sufficient statistics, N_{ijk}, requires explanation. The array is ordered first by increasing k, then increasing j, and then increasing i. Again, the variable ordering is as defined in the variable labels section of the file. One way to load the sufficient statistics into memory is to allocate an array of pointers to 2-dimensional matrices. There would be 6 matrices for the initial network and 9 matrices for the transition network. The dimensions of each matrix is $r_i \times q_i$, or the number of instantiations of the variable by the number of instantiations of the parents of the variable (see Appendix I).[3] The counts may be read directly into the matrices from the file, starting with the first column of the first variable to the last column of the last variable.

Instead of reading the sufficient statistics into an array of matrices stored in memory, one can reference the elements in the parameters file directly. For some specified variable i, parental instantiation j, and variable instantiation k, the value N_{ijk} is given by the following element in the list

$$k + r_i(j-1) + \sum_{i'=1}^{i-1} q_{i'} r_{i'}\,, \tag{A-1}$$

where q and r are as specified in Appendix I.[4]

It is important to clarify the ordering of the parental instantiations. If the variables $X_1, \ldots, X_n$ are instantiated to bins $b_1, \ldots, b_n$, the parental instantiation of variable X_i is given by

$$j = 1 + \sum_{i'=1}^{n} G(i', i)(b_{i'} - 1) \prod_{i''=1}^{i'-1} r_{i''}^{G(i'',i)}\,. \tag{A-2}$$

For example, suppose that a variable has three parents. The first parental instantiation will assign all parents to their first bin. The second parental instantiation will assign the first parent (as defined by the ordering in the variable labels portion of the file) to its second bin and the other two parents to their first bin. The sequence continues until all of the parents are instantiated to their last bins.

The following is a fragment of the parameter file. The lines that describe the sufficient statistics are truncated due to length.

```
# labels_initial
"A", "L", "v", "\dot v", "\dot h", "\dot \psi"
# G_initial
0 1 1 1 1 1
0 0 1 1 1 1
0 0 0 1 1 1
0 0 0 0 1 1
0 0 0 0 0 1
0 0 0 0 0 0
# r_initial
4 4 8 5 7 7
# N_initial
842712 970223 15706213 176025732 265497 315803 197208 64204 ...
# labels_transition
"A", "L", "v", "\dot v(t)", "\dot h(t)", "\dot \psi(t)", "\dot v(t+1)",
```

[3] For the transition network, note that the matrices for the variables that are not associated with time $t+1$ are empty.

[4] In the transition network, q_i is as defined in Appendix I for the nodes representing variables at time $t+1$. For the other nodes, i.e., the static variables and the variables at time t, q_i is set to zero.

```
"\dot h(t+1)", "\dot \psi(t+1)"
# G_transition
0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0 0
# r_transition
4 4 8 5 7 7 5 7 7
# N_transition
5 0 0 0 0 8 0 0 0 0 30 5 0 0 0 35 1 0 0 0 44 ...
# boundaries
*
*
0 30 60 90 120 140 165 250 300
-2 -1 -0.25 0.25 1 2
-2000 -1250 -750 -250 250 750 1250 2000
-8 -6 -4.5 -1.5 1.5 4.5 6 8
# resample_rates
0 0 0 0.019969 0.0394758 0.08752
```

This page intentionally left blank.

APPENDIX B
TRAJECTORY GENERATION

This appendix explains how to generate an encounter from the model. Software for parsing the parameters file (Appendix A) and generating trajectories is available from the authors.

B.1 INITIAL NETWORK SAMPLING

The first step in generating a random trajectory using the model is to sample from the Bayesian network representing the initial state distribution. To sample from a Bayesian network, as explained in Appendix I, one must first produce a topological sort of the nodes in the network. A topological sort orders the nodes of the network so that parents come before their descendants. The following is the graphical structure of the initial network as specified in the parameters file (see Appendix A) and shown in Figure B-1:

```
0 1 1 1 1 1
0 0 1 1 1 1
0 0 0 1 1 1
0 0 0 0 1 1
0 0 0 0 0 1
0 0 0 0 0 0
```

As can be seen, this ordering of the nodes is already topologically sorted: the first node (airspace class) is connected to all other nodes. The second node (altitude layer) is connected to all following nodes, and so on. The final node ($\dot{\psi}$) is not connected to any other nodes. We will see later that the transition network, in contrast, requires a topological sort.

With the nodes topologically sorted, we begin by sampling the first variable. As specified in the parameters file, the first variable is A, airspace class. Equation I-5, reproduced below, shows how to produce a random sample:

$$P(X_i = k \mid \pi_{ij}, D, G) = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i}(\alpha_{ijk'} + N_{ijk'})} \, .$$

In other words, the probability of selecting a particular airspace class is proportional to the prior (α_{ijk}) plus the frequency that airspace class appeared in the data (N_{ijk}). Following Cooper and Herskovits [20], we use an objective prior and set α_{ijk} to 1. To determine the values for N_{ijk}, we need to look at the sufficient statistics recorded in the parameters file. The sufficient statistics for the initial network is recorded as a long series of numbers. We use the scheme described in Appendix A to determine the actual values. These values turn out to be the first four numbers in the sufficient statistics sequence and are shown in the following table.

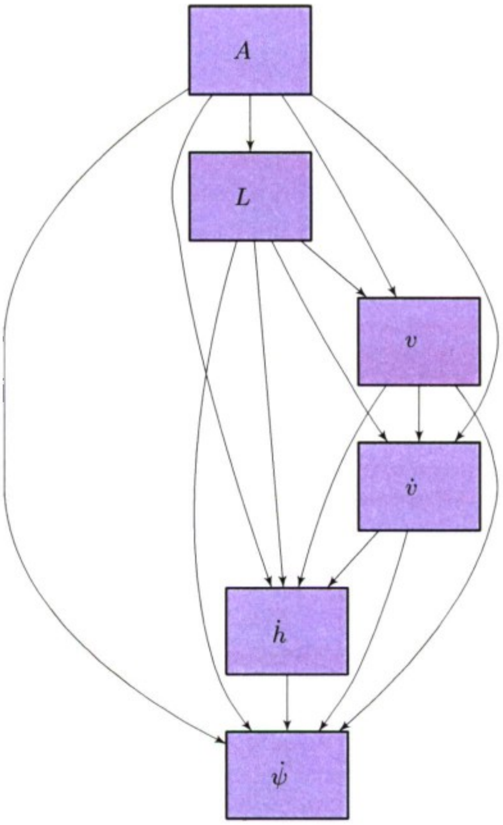


Figure B-1. A graphical representation of the initial network.

TABLE B-1

$N(A)$

A			
B	C	D	O
842712	970223	15706213	176025732

Now, we may compute the probability of selecting the different values of A using Equation I-5:

- $P(A = \mathrm{B}) = (842712 + 1)/(842712 + 1 + 970223 + 1 + 15706213 + 1 + 176025732 + 1) = 0.0044$
- $P(A = \mathrm{C}) = (970223 + 1)/(842712 + 1 + 970223 + 1 + 15706213 + 1 + 176025732 + 1) = 0.0050$
- $P(A = \mathrm{D}) = (15706213 + 1)/(842712 + 1 + 970223 + 1 + 15706213 + 1 + 176025732 + 1) = 0.0812$
- $P(A = \mathrm{O}) = (176025732 + 1)/(842712 + 1 + 970223 + 1 + 15706213 + 1 + 176025732 + 1) = 0.9095$

We use a random number generator to choose an airspace. For this example, let us choose O, the fourth instantiation of A.

The next step is to instantiate the second variable, L, which is altitude layer. Choosing a random instantiation for L is not quite as easy as for A because L depends upon other variables, namely A. We need to compute the conditional probability distribution $P(L \mid A = \mathrm{O})$, i.e., the distribution over the values of L given that A is O. We consult the sufficient statistics table $N(L \mid A)$ which is extracted from the parameters file (using the process explained in Appendix A) and is displayed in the table below.

TABLE B-2

$N(L \mid A)$

	L			
A	[500, 1200)	[1200, 3000)	[3000, 5000)	[5000, ∞)
B	265497	315803	197208	64204
C	500465	402396	67303	59
D	11209219	4496306	688	0
O	41043613	87491669	31033237	16457213

Since we are only interested in the counts associated with $A = \mathrm{O}$, we consider only the last row in the table. To translate the counts into probabilities, we add 1 to each element in the last row and then divide by the total resulting sum of that row. Again, we use a random number generator to select an airspace layer according to these probabilities. Suppose we choose layer $[500, 1200)$.

The next variable to be assigned is v. We consult the table for v, focusing on the row (highlighted below) that is consistent with the variable assignments so far, i.e., $A = \mathrm{O}$ and $L = [500, 1200)$. We choose a random airspeed based on the counts as with the previous variables.

TABLE B-3

$N(v \mid A, L)$

		v							
A	L	[0, 30)	[30, 60)	[60, 90)	[90, 120)	[120, 140)	[140, 165)	[165, 250)	[250, 300]
B	[500, 1200)	13503	31576	81977	95741	26229	10404	3795	2272
C	[500, 1200)	16282	32805	183761	197837	41048	18341	8069	2322
D	[500, 1200)	128191	583471	4645064	4180547	860838	447421	283816	79871
O	[500, 1200)	320032	1929263	13222694	17080198	5241637	2218202	890255	141332
B	[1200, 3000)	8673	12141	53304	123132	59683	39014	17051	2805
C	[1200, 3000)	8259	16204	97168	151460	56595	43606	26120	2984
D	[1200, 3000)	45303	110802	1193578	1951024	585532	333172	247795	29100
O	[1200, 3000)	218586	2516479	20054858	38148145	14552307	8113608	3645015	242671
B	[3000, 5000)	1494	9164	47800	77434	30430	17990	11636	1260
C	[3000, 5000)	1571	2705	24363	15223	9684	8597	4787	373
D	[3000, 5000)	0	38	159	278	137	22	26	28
O	[3000, 5000)	59902	822607	5610286	11325094	5793651	4435405	2843841	142451
B	[5000, 18000]	4094	4984	12720	19086	10292	6720	5553	755
C	[5000, 18000]	0	0	0	11	48	0	0	0
D	[5000, 18000]	0	0	0	0	0	0	0	0
O	[5000, 18000]	64081	185566	1421432	3514892	2983582	3611450	4160916	515294

This process of randomly assigning values to each of the variables conditional on the values of their parents continues until all of the variables have been assigned. For this example, assume that the following assignments have been made:

- $A = \mathrm{O}$
- $L = [500, 1200)$
- $v = [120, 140)$
- $\dot{v} = [-0.25, 0.25)$
- $\dot{h} = [-2000, -1250)$
- $\dot{\psi} = [-1.5, 1.5)$

Specific values within each bin are then determined based on a different uniform random number for each. In the case where a bin spans 0 (as it does in this example for $\dot{v}$, and $\dot{\psi}$), the value 0 itself is assigned.

B.2 TRANSITION NETWORK SAMPLING

We have described how to sample from the initial Bayesian network to generate a random initial state. The next step is to use the dynamic Bayesian network representing the transition distribution to generate the state at the next time step based on the initial state. The parameters file defines the following sequence of variables in the dynamic Bayesian network: $A, L, v, \dot{v}(t), \dot{h}(t), \dot{\psi}(t), \dot{v}(t+1), \dot{h}(t+1), \dot{\psi}(t+1)$. The parameters file also defines the following graphical structure for the network:

```
0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 1 1 1
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0 0
```

The graphical representation of this matrix is shown in Figure B-2. As can be seen, the ordering in the parameters file is not topologically sorted because there are arrows from $\dot{h}(t+1)$ and $\dot{\psi}(t+1)$ to $\dot{v}(t+1)$ but $\dot{v}(t+1)$ comes before both $\dot{h}(t+1)$ and $\dot{\psi}(t+1)$ in the list of variables in the parameters file. We need to topologically sort the list of variables to make sure that we assign values to variables in the right order. One possible topological sort involves simply moving $\dot{v}(t+1)$ to the end of the list. This sorting only affects the order in which the variables are sampled, not the semantics of the ordering in the parameters file.

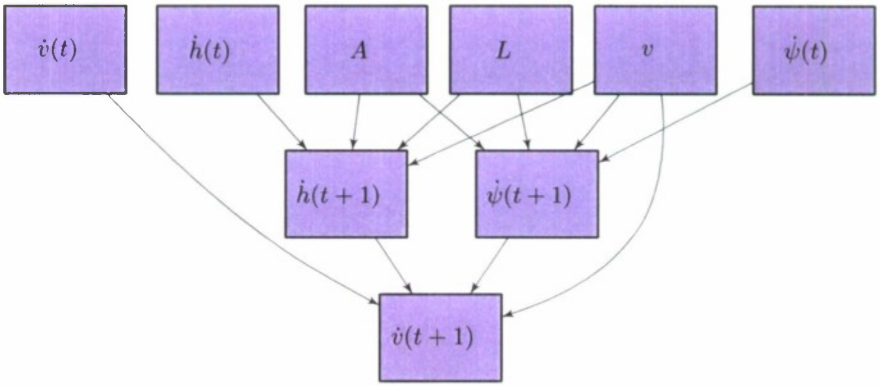


Figure B-2. A graphical representation of the transition network.

We are only interested in assigning new values to the dynamic variables, namely $\dot{h}(t+1)$, $\dot{\psi}(t+1)$ and $\dot{v}(t+1)$. The process is similar to the process used to sample from the initial network.

First, we consult the table of sufficient statistics for $\dot{h}(t+1)$. This table of conditional counts is shown below. For sampling purposes, we are only interested in the row that represents the current variable assignment, i.e., $A = \text{O}$, $L = [500, 1200)$, $v = [120, 140)$, $\dot{h} = [-2000, -1250)$, and $\dot{\psi} = [-1.5, 1.5)$. The relevant row is highlighted in Table B-4.

TABLE B-4

$$N(\dot{h}(t+1) \mid A, L, v, \dot{h}(t))$$

				$\dot{h}(t+1)$						
A	L	v	$\dot{h}(t)$	[−2000, −1250)	[−1250, −750)	[−750, −250)	[−250, 250)	[250, 750)	[750, 1250)	[1250, 2000]
B	[500, 1200)	[0, 30)	[−2000, −1250)	6	0	0	0	0	0	0
C	[500, 1200)	[0, 30)	[−2000, −1250)	46	3	0	0	0	0	0
D	[500, 1200)	[0, 30)	[−2000, −1250)	78	5	0	0	0	0	0
O	[500, 1200)	[0, 30)	[−2000, −1250)	333	7	0	0	0	0	0
B	[1200, 3000)	[0, 30)	[−2000, −1250)	39	0	0	0	0	0	0
C	[1200, 3000)	[0, 30)	[−2000, −1250)	196	1	0	0	0	0	0
D	[1200, 3000)	[0, 30)	[−2000, −1250)	132	2	0	0	0	0	0
O	[1200, 3000)	[0, 30)	[−2000, −1250)	638	20	0	0	0	0	0
B	[3000, 5000)	[0, 30)	[−2000, −1250)	94	0	0	0	0	0	0
C	[3000, 5000)	[0, 30)	[−2000, −1250)	90	0	0	0	0	0	0
D	[3000, 5000)	[0, 30)	[−2000, −1250)	0	0	0	0	0	0	0
O	[3000, 5000)	[0, 30)	[−2000, −1250)	347	12	0	0	0	0	0
B	[5000, 18000]	[0, 30)	[−2000, −1250)	160	2	0	0	0	0	0
C	[5000, 18000]	[0, 30)	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[0, 30)	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[0, 30)	[−2000, −1250)	1359	21	1	0	0	0	0
B	[500, 1200)	[30, 60)	[−2000, −1250)	178	7	0	0	0	0	0
C	[500, 1200)	[30, 60)	[−2000, −1250)	34	3	0	0	0	0	0
D	[500, 1200)	[30, 60)	[−2000, −1250)	1178	46	0	0	0	0	0
O	[500, 1200)	[30, 60)	[−2000, −1250)	3805	250	0	0	0	0	0
B	[1200, 3000)	[30, 60)	[−2000, −1250)	113	0	0	0	0	0	0
C	[1200, 3000)	[30, 60)	[−2000, −1250)	90	4	0	0	0	0	0
D	[1200, 3000)	[30, 60)	[−2000, −1250)	617	9	0	0	0	0	0
O	[1200, 3000)	[30, 60)	[−2000, −1250)	9425	390	0	0	0	0	0
B	[3000, 5000)	[30, 60)	[−2000, −1250)	198	3	0	0	0	0	0
C	[3000, 5000)	[30, 60)	[−2000, −1250)	27	2	0	0	0	0	0
D	[3000, 5000)	[30, 60)	[−2000, −1250)	0	0	0	0	0	0	0
O	[3000, 5000)	[30, 60)	[−2000, −1250)	6431	201	0	0	0	0	0
B	[5000, 18000]	[30, 60)	[−2000, −1250)	398	1	0	0	0	0	0
C	[5000, 18000]	[30, 60)	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[30, 60)	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[30, 60)	[−2000, −1250)	5530	68	0	0	0	0	0
B	[500, 1200)	[60, 90)	[−2000, −1250)	258	14	0	0	0	0	0
C	[500, 1200)	[60, 90)	[−2000, −1250)	338	5	0	0	0	0	0
D	[500, 1200)	[60, 90)	[−2000, −1250)	3747	178	0	0	0	0	0
O	[500, 1200)	[60, 90)	[−2000, −1250)	8469	593	0	0	0	0	0
B	[1200, 3000)	[60, 90)	[−2000, −1250)	341	5	0	0	0	0	0
C	[1200, 3000)	[60, 90)	[−2000, −1250)	293	5	0	0	0	0	0
D	[1200, 3000)	[60, 90)	[−2000, −1250)	2057	68	0	0	0	0	0
O	[1200, 3000)	[60, 90)	[−2000, −1250)	29798	1328	1	0	0	0	0
B	[3000, 5000)	[60, 90)	[−2000, −1250)	684	11	0	0	0	0	0
C	[3000, 5000)	[60, 90)	[−2000, −1250)	62	4	0	0	0	0	0
D	[3000, 5000)	[60, 90)	[−2000, −1250)	0	0	0	0	0	0	0
O	[3000, 5000)	[60, 90)	[−2000, −1250)	22160	710	2	0	0	0	0
B	[5000, 18000]	[60, 90)	[−2000, −1250)	915	3	0	0	0	0	0
C	[5000, 18000]	[60, 90)	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[60, 90)	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[60, 90)	[−2000, −1250)	12667	231	0	0	0	0	0
B	[500, 1200)	[90, 120)	[−2000, −1250)	196	5	0	0	0	0	0

Continued on next page. . .

TABLE B-4. Continued

				$\dot{h}(t+1)$						
A	L	v	$\dot{h}(t)$	[−2000, −1250)	[−1250, −750)	[−750, −250)	[−250, 250)	[250, 750)	[750, 1250)	[1250, 2000]
C	[500, 1200)	[90, 120)	[−2000, −1250)	116	6	0	0	0	0	0
D	[500, 1200)	[90, 120)	[−2000, −1250)	3228	232	0	0	0	0	0
O	[500, 1200)	[90, 120)	[−2000, −1250)	8398	748	2	0	0	0	0
B	[1200, 3000)	[90, 120)	[−2000, −1250)	291	3	0	0	0	0	0
C	[1200, 3000)	[90, 120)	[−2000, −1250)	388	13	0	0	0	0	0
D	[1200, 3000)	[90, 120)	[−2000, −1250)	4455	151	1	0	0	0	0
O	[1200, 3000)	[90, 120)	[−2000, −1250)	55955	2344	1	0	0	0	0
B	[3000, 5000)	[90, 120)	[−2000, −1250)	995	14	0	0	0	0	0
C	[3000, 5000)	[90, 120)	[−2000, −1250)	68	4	0	0	0	0	0
D	[3000, 5000)	[90, 120)	[−2000, −1250)	0	0	0	0	0	0	0
O	[3000, 5000)	[90, 120)	[−2000, −1250)	39500	971	0	0	0	0	0
B	[5000, 18000]	[90, 120)	[−2000, −1250)	954	12	0	0	0	0	0
C	[5000, 18000]	[90, 120)	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[90, 120)	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[90, 120)	[−2000, −1250)	24326	445	0	0	0	0	0
B	[500, 1200)	[120, 140)	[−2000, −1250)	41	3	0	0	0	0	0
C	[500, 1200)	[120, 140)	[−2000, −1250)	48	4	0	0	0	0	0
D	[500, 1200)	[120, 140)	[−2000, −1250)	1673	136	0	0	0	0	0
O	[500, 1200)	[120, 140)	[−2000, −1250)	4540	347	1	0	0	0	0
B	[1200, 3000)	[120, 140)	[−2000, −1250)	189	0	0	0	0	0	0
C	[1200, 3000)	[120, 140)	[−2000, −1250)	372	15	0	0	0	0	0
D	[1200, 3000)	[120, 140)	[−2000, −1250)	3280	95	0	0	0	0	0
O	[1200, 3000)	[120, 140)	[−2000, −1250)	31536	1262	2	0	0	0	0
B	[3000, 5000)	[120, 140)	[−2000, −1250)	484	7	0	0	0	0	0
C	[3000, 5000)	[120, 140)	[−2000, −1250)	195	5	0	0	0	0	0
D	[3000, 5000)	[120, 140)	[−2000, −1250)	0	0	0	0	0	0	0
O	[3000, 5000)	[120, 140)	[−2000, −1250)	23949	483	1	0	0	0	0
B	[5000, 18000]	[120, 140)	[−2000, −1250)	586	3	0	0	0	0	0
C	[5000, 18000]	[120, 140)	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[120, 140)	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[120, 140)	[−2000, −1250)	16348	253	0	0	0	0	0
B	[500, 1200)	[140, 165)	[−2000, −1250)	18	3	0	0	0	0	0
C	[500, 1200)	[140, 165)	[−2000, −1250)	39	2	0	0	0	0	0
D	[500, 1200)	[140, 165)	[−2000, −1250)	1555	108	0	0	0	0	0
O	[500, 1200)	[140, 165)	[−2000, −1250)	3411	264	0	0	0	0	0
B	[1200, 3000)	[140, 165)	[−2000, −1250)	229	6	0	0	0	0	0
C	[1200, 3000)	[140, 165)	[−2000, −1250)	192	4	0	0	0	0	0
D	[1200, 3000)	[140, 165)	[−2000, −1250)	2799	82	0	0	0	0	0
O	[1200, 3000)	[140, 165)	[−2000, −1250)	23885	904	0	0	0	0	0
B	[3000, 5000)	[140, 165)	[−2000, −1250)	250	4	0	0	0	0	0
C	[3000, 5000)	[140, 165)	[−2000, −1250)	81	1	0	0	0	0	0
D	[3000, 5000)	[140, 165)	[−2000, −1250)	0	0	0	0	0	0	0
O	[3000, 5000)	[140, 165)	[−2000, −1250)	21005	434	0	0	0	0	0
B	[5000, 18000]	[140, 165)	[−2000, −1250)	312	3	0	0	0	0	0
C	[5000, 18000]	[140, 165)	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[140, 165)	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[140, 165)	[−2000, −1250)	18853	389	0	0	0	0	0
B	[500, 1200)	[165, 250)	[−2000, −1250)	21	0	0	0	0	0	0
C	[500, 1200)	[165, 250)	[−2000, −1250)	33	4	0	0	0	0	0
D	[500, 1200)	[165, 250)	[−2000, −1250)	2091	143	0	0	0	0	0
O	[500, 1200)	[165, 250)	[−2000, −1250)	3920	292	0	0	0	0	0
B	[1200, 3000)	[165, 250)	[−2000, −1250)	171	3	0	0	0	0	0
C	[1200, 3000)	[165, 250)	[−2000, −1250)	313	12	0	0	0	0	0
D	[1200, 3000)	[165, 250)	[−2000, −1250)	3528	90	0	0	0	0	0

Continued on next page. . .

TABLE B-4. Continued

				$\dot{h}(t+1)$						
A	L	v	$\dot{h}(t)$	[−2000, −1250)	[−1250, −750)	[−750, −250)	[−250, 250)	[250, 750)	[750, 1250)	[1250, 2000]
O	[1200, 3000)	[165, 250)	[−2000, −1250)	36915	1283	0	0	0	0	0
B	[3000, 5000)	[165, 250)	[−2000, −1250)	293	5	0	0	0	0	0
C	[3000, 5000)	[165, 250)	[−2000, −1250)	275	1	0	0	0	0	0
D	[3000, 5000)	[165, 250)	[−2000, −1250)	6	0	0	0	0	0	0
O	[3000, 5000)	[165, 250)	[−2000, −1250)	45070	794	1	0	0	0	0
B	[5000, 18000]	[165, 250)	[−2000, −1250)	133	0	0	0	0	0	0
C	[5000, 18000]	[165, 250)	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[165, 250)	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[165, 250)	[−2000, −1250)	59358	779	3	0	0	0	0
B	[500, 1200)	[250, 300]	[−2000, −1250)	188	4	0	0	0	0	0
C	[500, 1200)	[250, 300]	[−2000, −1250)	13	0	0	0	0	0	0
D	[500, 1200)	[250, 300]	[−2000, −1250)	1693	83	0	0	0	0	0
O	[500, 1200)	[250, 300]	[−2000, −1250)	2663	132	0	0	0	0	0
B	[1200, 3000)	[250, 300]	[−2000, −1250)	325	6	0	0	0	0	0
C	[1200, 3000)	[250, 300]	[−2000, −1250)	178	3	0	0	0	0	0
D	[1200, 3000)	[250, 300]	[−2000, −1250)	2199	39	0	0	0	0	0
O	[1200, 3000)	[250, 300]	[−2000, −1250)	14756	301	0	0	0	0	0
B	[3000, 5000)	[250, 300]	[−2000, −1250)	143	0	0	0	0	0	0
C	[3000, 5000)	[250, 300]	[−2000, −1250)	43	1	0	0	0	0	0
D	[3000, 5000)	[250, 300]	[−2000, −1250)	5	0	0	0	0	0	0
O	[3000, 5000)	[250, 300]	[−2000, −1250)	19204	222	0	0	0	0	0
B	[5000, 18000]	[250, 300]	[−2000, −1250)	228	2	0	0	0	0	0
C	[5000, 18000]	[250, 300]	[−2000, −1250)	0	0	0	0	0	0	0
D	[5000, 18000]	[250, 300]	[−2000, −1250)	0	0	0	0	0	0	0
O	[5000, 18000]	[250, 300]	[−2000, −1250)	54066	522	1	0	0	0	0

We look at the row of interest and randomly assign a value to $\dot{h}(t+1)$ with probability proportional to the corresponding elements plus 1. We then assign $\dot{\psi}(t+1)$ to a random value conditional on A, L, $\dot{\psi}(t)$, and v. Finally, we assign $\dot{v}(t+1)$ based on the assignments of v, $\dot{v}(t)$, $\dot{h}(t+1)$, and $\dot{\psi}(t+1)$. The process is repeated for each time step. The length of the trajectory may be made as long as desired.

Sampling from the Bayesian networks produces a sequence of assignments of variables to bins, e.g., $[250, 750)$. As Section 4 describes, the next step is to sample uniformly within the bins to produce real values (with the exception of a bin spanning 0 in which case 0 itself is selected). If we simply sample within each bin at each time step, there would be excessive variability in the vertical rates, turn rates, and acceleration. We therefore only resample within bins at mean rates specified in the parameters file:

```
0 0 0 0.019969 0.0394758 0.08752
```

The first three variables have zero as their rates because they do not change during the course of the trajectory. The last three elements specify the rates with which $\dot{v}$, $\dot{h}$, and $\dot{\psi}$ change within bins. As the trajectory evolves, whenever a variable switches to a new bin we must sample within the bin. Typically, the values of variables continue in the same bin at the next time step. If the bin remains the same, we either continue with the same value or resample within the bin according to the specified rates.

Once the initial conditions and a series of control variables ($\dot{v}$, $\dot{h}$, and $\dot{\psi}$) have been selected, the aircraft trajectory is constructed or simulated using an appropriate dynamic model. The control variables are assumed to be held constant across each 1 s time step.

B.3 VALIDATION

One way to validate an implementation of the sampling scheme is to generate a large collection of samples and compare the generated feature distributions against the distributions shown in Appendix C for the full dataset used to construct the model (shown in black in Figures C-1 and C-2). Since the histograms only show the marginal distributions for the variables and does not reveal the important correlation between variables, an approach similar to that discussed in Appendix C is appropriate. So long as the ratio in Equation C-1 is large, then one may be confident that the sampling scheme was implemented correctly. When we generated 1,000 samples, we found that the natural logarithm of the ratio was 1.4×10^3 for the initial network and 1.5×10^3 for the transition network, indicating with extremely high confidence that our 1,000 samples came from the same distribution as the model.

APPENDIX C
SEASONAL AND GEOGRAPHIC COMPARISON

The model described in this report represents the average behavior of 1200-squawking aircraft in the continental United States. If this model is to be used for more focused safety studies, for example studies concerned with only a particular region, it is important to understand how the behavior of a smaller population compares to the national average.

We found almost no seasonal variation of trajectory characteristics in the radar data. Figure C-1 shows the distributions over the features that define the model for the full dataset and compares them with data from December and June. Although the behavior of the aircraft across seasons are similar, the variation of traffic densities do vary significantly. Because VFR traffic are restricted to visual meteorological conditions, the density of VFR traffic is strongly influenced by weather. In areas with fairly consistent visual conditions, such as California, the traffic density is fairly constant, but in areas with varying weather condition, such the Northeast, the traffic density fluctuates as one would expect.

To determine how traffic behavior changes with location, we compared the feature distributions for traffic captured by a Los Angeles, California sensor and a Manchester, New Hampshire sensor. As Figure C-2 shows, the distributions over airspace class and altitude layer are significantly different from the national average. This difference is to be expected because both the Los Angeles and Manchester sensors are located in terminal areas. There are also significant differences in airspeed. The acceleration, vertical rate, and turn rate distributions are more comparable.

We hypothesized that the geographic variation in airspeed, acceleration, vertical rate, and turn rate was due to differences in the distribution over airspace class and altitude layer. To test this hypothesis, we generated new distributions from the national data, but controlling for airspace class and altitude layer. We used the Bayesian network parameters estimated from the national data and substituted the airspace class and altitude layer parameters estimated from the Los Angeles data. We sampled from the resulting joint distribution to estimate the marginal distributions for the airspeed, acceleration, vertical rate, and turn rate variables. If the differences in the feature distributions were due only to the differences in airspace class and altitude layer, then the sampled marginal distributions would resemble the distributions from the Los Angeles data. As Figure C-3 shows, the acceleration, vertical rate, and turn rate distributions match, but the airspeed distribution remains somewhat different. The airspeed near terminal areas, such as Los Angeles and Manchester, seem to be somewhat slower than the national average would predict, even when compensated for airspace class and altitude. If a safety study is to focus specifically on a terminal area, it may be worth examining how the distribution in airspeeds differ from the national average and adjust the model accordingly.

The plots of the marginal distributions of the various features provide a visual way of comparing the airspace properties of different seasons and geographic regions. The marginal distribution plots do not capture, however, the relationships (correlations) between variables, which is especially important when comparing the dynamic behavior of aircraft. What we wish to determine is the probability that two different populations, e.g., Los Angeles and the national average, have the same Bayesian network parameters given the recorded data (assuming that the graphical structure of both networks are identical). Let us denote the true parameters of the two populations as $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and the data associated with the two populations as D_1 and D_2. We wish to compute the posterior $P(\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 \mid D_1, D_2)$, which involves multiplying the prior distribution of the hypothesis that $P(\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2)$ by the ratio

$$\frac{P(D_1, D_2 \mid \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2)}{P(D_1, D_2)} = \frac{\int P(D_1, D_2 \mid \boldsymbol{\theta})p(\boldsymbol{\theta})\, d\boldsymbol{\theta}}{\left[\int P(D_1 \mid \boldsymbol{\theta})p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right]\left[\int P(D_2 \mid \boldsymbol{\theta})p(\boldsymbol{\theta})\, d\boldsymbol{\theta}\right]}. \tag{C-1}$$

If we define $f(D) = \int P(D|\boldsymbol{\theta})p(\boldsymbol{\theta})\, d\boldsymbol{\theta}$, the ratio becomes

$$\frac{f(D_1 \cup D_2)}{f(D_1)f(D_2)}.$$

As shown by Cooper and Herskovits [20] and using the notation used in Appendix I,

$$f(D) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}.$$

We computed the natural logarithm of the ratio in Equation C-1 to compare the feature distributions from different data sets (see Table C-1). The more positive the log-ratio is, the more likely the distributions are the same. Negative log-ratios indicate that the distributions are different. The transition distributions for different seasons and different geographic locations were found to be statistically similar. We conclude that although there is spatial and temporal variation in traffic density and spatial variation in airspace class, altitude layer, and airspeed, the evolution of trajectories follow a very similar stochastic process conditioned on airspace class, altitude layer, and airspeed.

TABLE C-1

Log ratio comparing the transition distribution of different populations.

Population 1	Population 2	Log Ratio
1–7 December 2008	1–7 June 2008	6.3×10^4
Manchester, NH	Los Angeles, CA	2.3×10^4
Manchester, NH	National Average	2.6×10^4
Los Angeles, CA	National Average	3.8×10^4

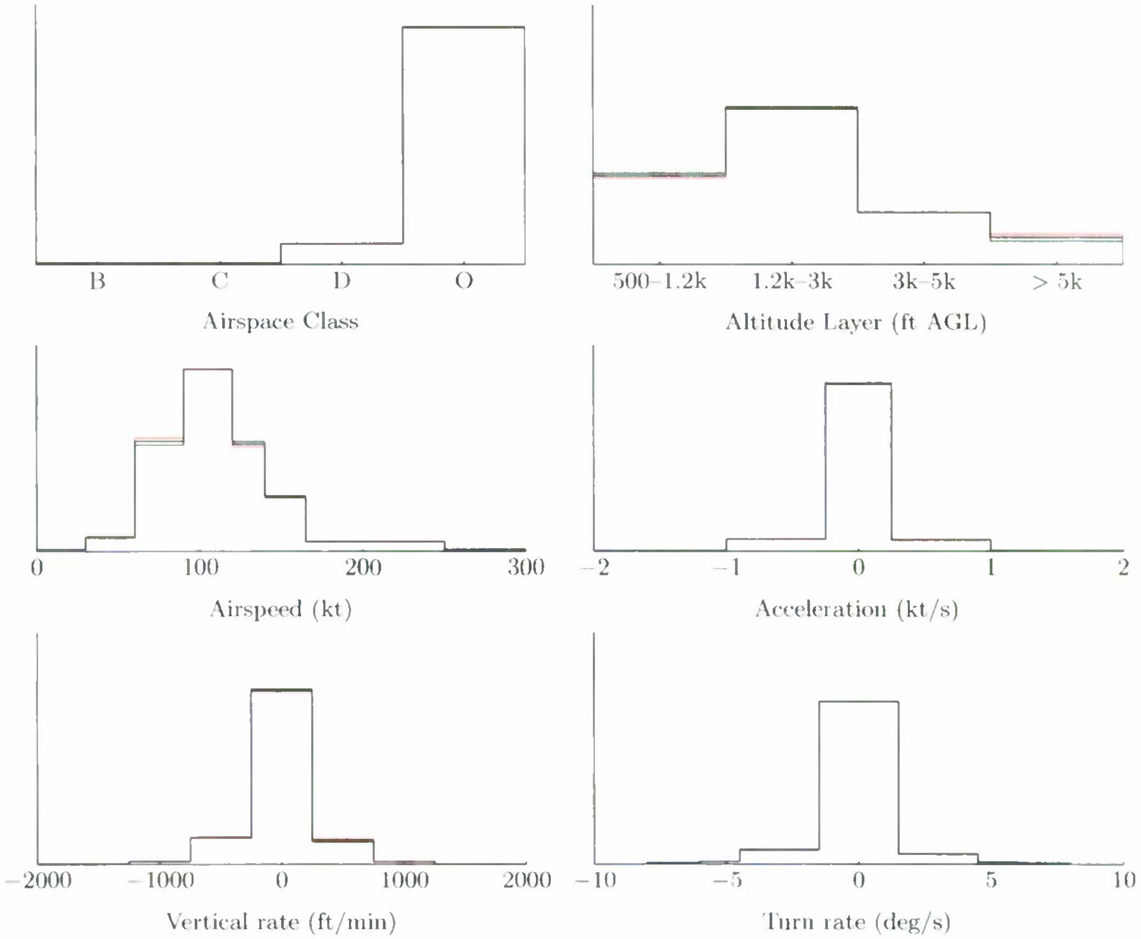


Figure C-1. Seasonal feature distribution comparison. Shown in black are the distribution of features from the dataset used to create the model, which came from recorded radar feeds from 1–7 December 2007 and 1–7 June 2008. Shown in green are the feature distributions from only 1–7 December 2007. Shown in red are the feature distributions from only 1–7 June 2008. As the plots reveal, the distributions for December and June are nearly identical.

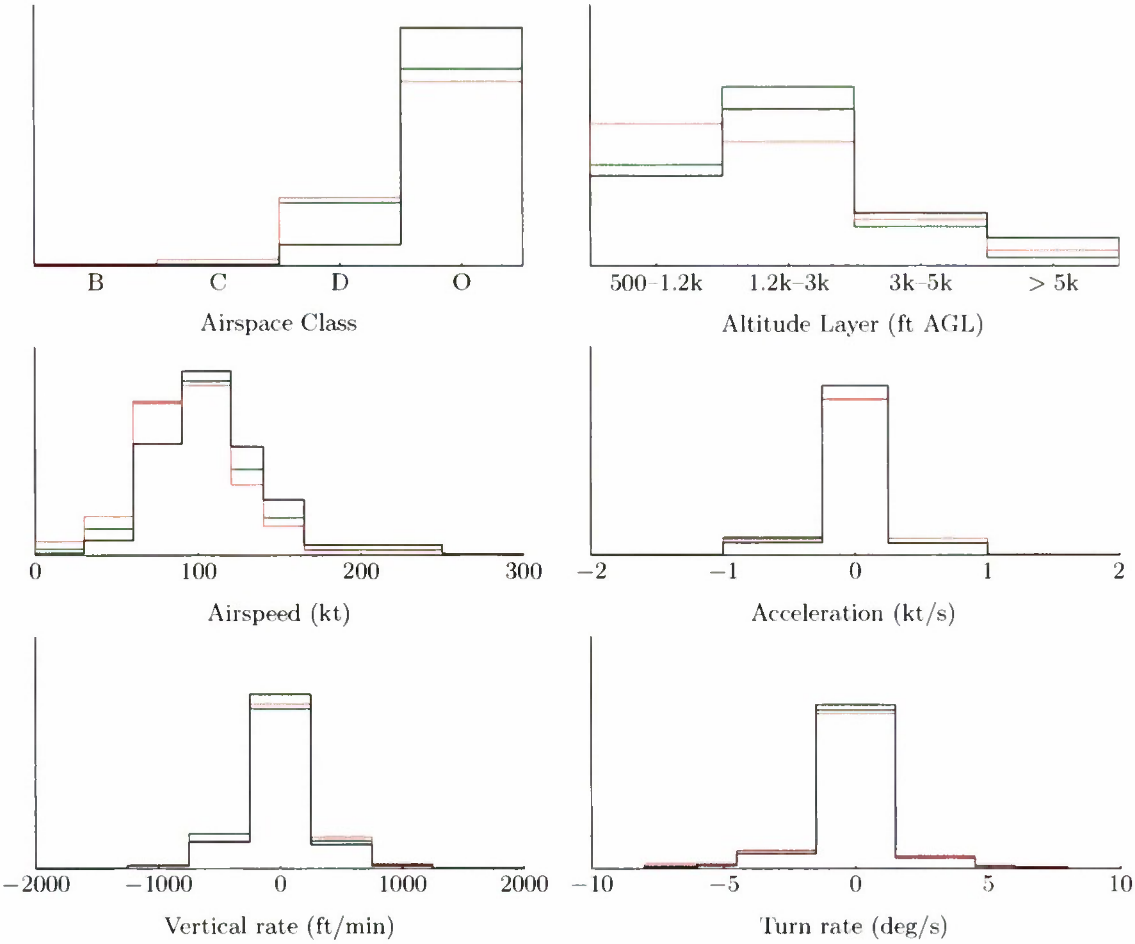


Figure C-2. Geographical feature distribution comparison. Shown in black are the distribution of features from the dataset used to create the model. Shown in red are the feature distributions from the LAXB sensor located in in Los Angeles, California. Shown in green are the feature distributions from MHT located in Manchester, New Hampshire.

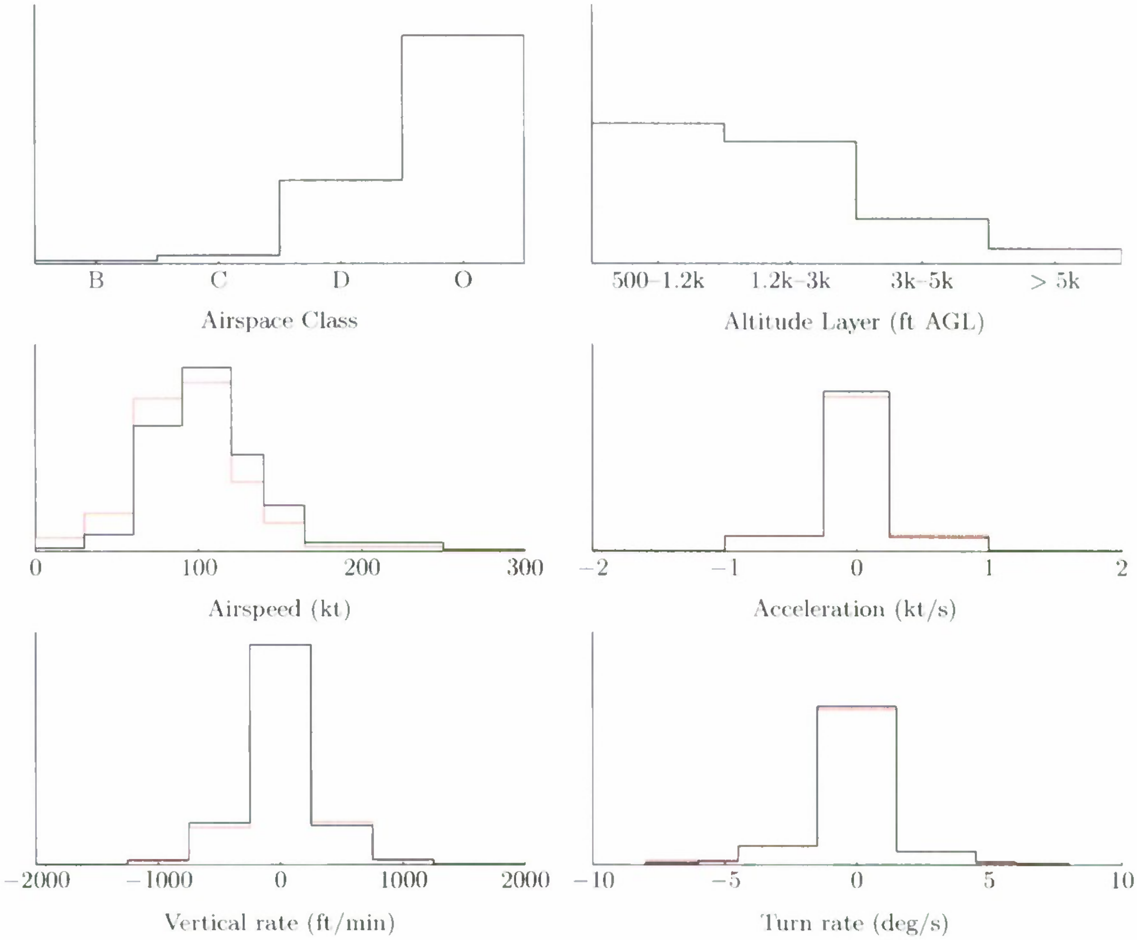


Figure C-3. Geographical feature distribution comparison corrected for airspace class and altitude layer. Shown in red are the feature distributions from the LAXB sensor located in in Los Angeles, California. Shown in black are the feature distributions from the national model but adjusted to reflect the airspace class and altitude layer distributions in Los Angeles.

This page intentionally left blank.

APPENDIX D
NONCOOPERATIVE TRAFFIC COMPARISON

Modeling noncooperative traffic is especially important when evaluating collision avoidance systems for unmanned aircraft. Unmanned aircraft typically rely upon electro-optical or primary radar sensors to detect aircraft without transponders. This report described a general framework that can be used for constructing an uncorrelated encounter model from *any* data source, but the model was based on cooperative 1200-code tracks captured by secondary surveillance radar. A key question is whether such a model, based on cooperative aircraft, can be used as a surrogate for noncooperative aircraft. Many noncooperative aircraft are single-engine fixed-wing airplanes similar to those that fly cooperatively using 1200-codes. Thus, modeling 1200-code characteristics will include these conventional types of noncooperative aircraft. On the other hand, there are also noncooperative aircraft that typically do not use transponders (e.g., balloons) and so would not be represented in the model described in this report.

Constructing a model based on primary-only radar returns from aircraft without transponders is significantly more challenging, as discussed in the introduction of this report. Of the generally-available surveillance radars that have height-finding capabilities, height estimates are typically quite poor, making it difficult to infer altitude and vertical rates. Another problem with primary-only data is that signficant clutter can disrupt the development of clean aircraft tracks needed to extract features such as turn rate.

To assess the degree to which the model reported here can be used as a surrogate for some noncooperative aircraft, we built specialized primary-only target classifiers. These classifiers take as input radar track coordinates and output whether it is likely that the track represents an aircraft (as opposed to a flock of birds or ground clutter, for example). Based on a collection of recorded radar data from 8 terminal radar sensors, we developed Gaussian mixture models representing the class-conditional distributions. The details of this analysis are discussed elsewhere [21].

Two classifiers were implemented. The first classifier was constructed using supervised learning techniques and based on two data sets: (1) a set of known non-aircraft primary-only tracks; and (2) a set of 1200-code cooperative aircraft tracks. The non-aircraft tracks consisted of primary-only tracks within class B airspace, where a transponder is almost always required. Although it is possible that some aircraft without transponders were operating in class B airspace (e.g., during formation flight), we are confident that the vast majority of the tracks observed within class B airspace are due to birds and other non-aircraft phenomena. The second data set based on 1200-code tracks were obtained by filtering for the appropriate Mode A code from the sensors used in the analysis. The two data sets, then, provided mutually-exclusive examples of cases known with confidence to be either (1) non-aircraft tracks or (2) aircraft tracks.

The second classifier was constructed using semi-supervised learning techniques. Training sets in this case included: (1) a set of known non-aircraft primary-only tracks similar to the prior case; and (2) a large collection of unlabeled (unknown) primary-only tracks.

Sets of primary-only tracks were then passed into these classifiers, which in turn output a three-way decision: the track was either (1) "conventional": an aircraft similar to a 1200-code aircraft; (2) "unconventional": an aircraft not similar to a 1200-code aircraft; or (3) not an aircraft. By appropriately tuning the classifiers using test tracks, it was possible to obtain correct classifications at a rate above 90%.

Figure D-1 compares feature histograms from 1200-code beacon tracks against the noncooperative aircraft tracks identified by the two classifiers. Cooperative aircraft features are shown in black, representing typical 1200-code conventional aircraft behavior. Primary-only tracks classified as "conventional" are shown in red; primary-only tracks classified as "unconventional" are shown in green. Not shown are the characteristics of tracks identified as "not aircraft". For reference, a distribution of glider airspeeds is also provided in cyan, extracted from publicly-available repositories of recorded GPS tracks.[5]

As shown, the airspeeds, accelerations, and turn rates of "conventional" primary-only tracks are comparable to cooperative 1200-code aircraft. Thus, there is a population of noncooperative aircraft that are considered to be adequately captured by the uncorrelated model described in this document. The airspeeds of "unconventional" primary-only tracks are significantly lower than 1200-code aircraft, and in addition have somewhat lower airspeed acceleration and more frequent non-zero turn rates; the model presented here is therefore not considered to provide an adequate representation of characteristics for some classes of aircraft.

Gliders are likely to represent a signification fraction of the "unconventional" noncooperative aircraft, and there are other categories of unconventional aircraft such as balloons (both manned and unmanned) and powered ultralights. Separate encounter models are currently being developed specifically for these unconventional aircraft.

To summarize, features from a certain class of noncooperative aircraft are adequately captured by the uncorrelated model presented in this document. The advantage of using this encounter model, derived from cooperative 1200-code aircraft, is that the aircraft tracks are reliable and contain altitude information. The result is greater confidence that appropriate aircraft dynamics are modeled, and it is therefore advantageous to use this model as a surrogate for many noncooperative aircraft. For those aircraft that behave in a more unconventional manner, such as gliders and balloons, a separate model is required and being developed.

A more focused analysis of a particular operation region would require first determining the relative populations of "conventional" and "unconventional" noncooperative aircraft using the classifiers developed here operating on primary-only radar data. Then, collision risk estimates from each of the appropriate models would be combined in a weighted manner to provide an overall, adjusted risk level based on that traffic mix.

[5] See http://soaringweb.org and http://www.paraglidingforum.com/modules.php?name=leonardo&op=list_flights.

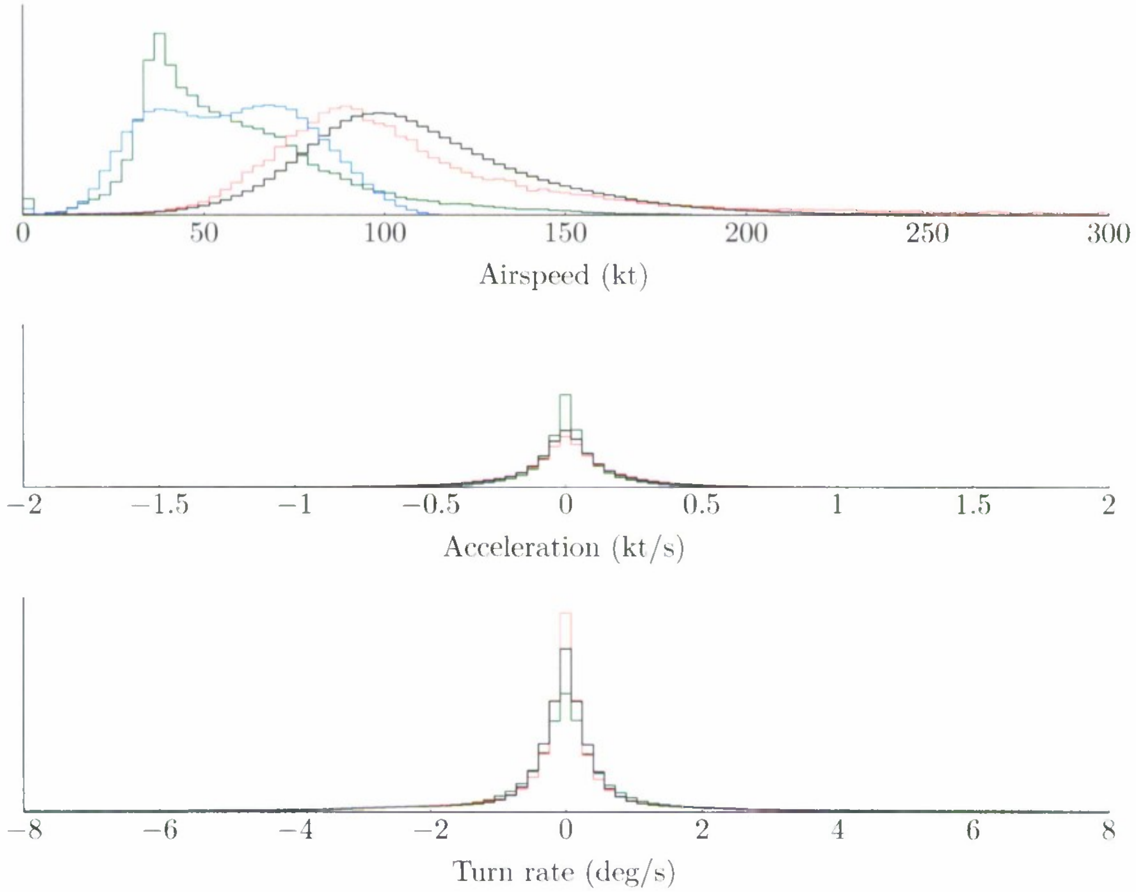


Figure D-1. Feature histograms for cooperative 1200-code aircraft (black), conventional noncooperative aircraft (red), unconventional noncooperative aircraft (green), and known glider aircraft (cyan, airspeed only). Vertical rates cannot be inferred from primary only tracks.

This page intentionally left blank.

APPENDIX E
DYNAMIC SIMULATION VALIDATION

This appendix describes a set of sample trajectories that may be used as validation for those wishing to implement their own dynamic simulation models. A text file, named `uncor_tracks.txt` and available from Lincoln Laboratory, contains 50 synthetic tracks. Each track is 50 seconds in length and was generated by sampling model parameters from the dynamic Bayesian network and then simulating the aircraft trajectory in Lincoln Laboratory's dynamic simulation model.

The text file is a space-delimited list with the following ten columns:

- **ID:** Each row begins with an `id` number. All rows with the same `id` number correspond to a single track.
- **Time t:** Time values are reported once per second in the text file.
- **Vertical rate $\dot{h}$:** The vertical rate is reported in ft/min. This value is sampled from the dynamic Bayesian network.
- **Airspeed acceleration $\dot{v}$:** The airspeed acceleration is reported in kt/s and is also sampled from the dynamic Bayesian network.
- **Turn rate $\dot{\psi}$:** Turn rate is reported in deg/s and is also a variable from the dynamic Bayesian network.
- **Airspeed v:** The initial airspeed (kt) is defined by sampling from the initial Bayesian network. The remaining values for $t > 0$ are outputs from Lincoln Laboratory's dynamic simulation.
- **North position x:** Each aircraft is initialized at $x = 0$ ft.
- **East position y:** Each aircraft is initialized at $y = 0$ ft.
- **Altitude h:** All sample aircraft trajectories are initialized at an altitude of 10,000 ft.
- **Heading ψ:** All aircraft are initialized heading due north ($\psi = 0$ deg).

The dynamic variables ($\dot{h}$, $\dot{v}$, and $\dot{\psi}$) are inputs to an aircraft dynamic model that describe how the aircraft maneuvers throughout the simulation. The variables v, x, y, and h can be used to compare outputs of a simulation. If an aircraft is initialized at the origin with zero heading and an altitude of 10,000 ft, then the simulated track should approximately match the values provided in the text file.

Exactly matching simulation outputs with values in the text file is extremely unlikely due to the variety of acceptable aircraft models in simulation (e.g., degrees-of-freedom, transient dynamics, simulation step size, etc). The simulated and provided tracks only need to approximate each other in both the vertical and horizontal plane. Note that small differences in heading, resulting from a turn early in the simulation, can result in large positional deviations further into the simulation—horizontal errors resulting from small differences in aircraft headings are acceptable for validating an aircraft dynamic simulation model and implementation of the uncorrelated encounter model.

Since there are numerous aircraft models that may be correctly implemented in simulation, this appendix does not define thresholds or minimum errors for validating a simulation. Instead, we only require that simulated aircraft must track steady-state inputs. For example, if sampling from the dynamic Bayesian network results in an aircraft flying straight (i.e., $\dot{\psi} = 0$ deg/s) that later transitions to $\dot{\psi} = 2$ deg/s, then the aircraft must eventually turn clockwise at 2 deg/s—the transient between $\dot{\psi} = 0$ and $\dot{\psi} = 2$ can differ from the sample trajectories.

APPENDIX F
TRACKING AND FUSION

Converting raw radar reports into tracks that are usable for our model development is a two-stage process. The first stage involves forming local tracks from the reports associated with each sensor. The second stage involves fusing the local tracks from multiple sensors to form global tracks. This appendix provides a brief overview of tracking and fusion.

We use the tracking algorithms from the Mode S and ASR-9 systems [13], the two most modern sensors in the Air Traffic Control System. The beacon correlation algorithms come from Mode S and the primary radar correlation algorithms come from ASR-9 Processor Augmentation Card (9-PAC). Both systems are integrated to provide a consistent track, although for the purpose of this report we ignore primary only reports.

After the reports of each sensor have been correlated into local tracks, we can fuse the local tracks to provide a global picture of the airspace. Fusion performs the following functions:

1. Merge tracks from multiple sensors that correspond to the same aircraft into a single global track.

2. Compute the speed and heading of each global track to permit trajectory predictions.

3. Correct sensor tracking errors that would have led to split global tracks and thus false encounters.

We use a *track-to-track* fusion method, meaning that we track each sensor's individual reports and then we merge all of the tracks [22]. The main advantages of *track-to-track* fusion over *report-to-track* fusion, in which all reports are correlated directly to global tracks, are:

1. Bias independence for velocity determination and maneuver detection.

2. Removal of short update interval velocity anomalies.

3. Reduced likelihood of forming clutter tracks.

4. Reduced likelihood of introducing incorrect data points into the track due to correlation errors.

Track merging employs position, velocity, Mode A code, and altitude as matching attributes over the entire track. Track merging for tracks with discrete codes, which are unique within an area, employs large correlation boxes for each of the matching attributes. Other tracks (1200 code or radar-only) must pass more stringent position tests and velocity tests in order to be merged together.

Our fusion method works forward in time. Thus, there is often doubt in whether or not tracks from multiple radars are indeed the same track with just a few data points. In cases of doubt, tentative matches are remembered that can be upgraded to a merge after more scans of information are obtained. Merges are checked each scan and can be undone if later found to be unsatisfactory. Aircraft code changes are also accommodated, although they must be verified by other sensors in the merge set of the global track before being accepted. If only one sensor reports a code change, it is assumed that the sensor had a track swap, and the merge situation is altered accordingly.

The remainder of this section explains how we add local sensor tracks to global tracks, how to estimate track velocity as part of the fusion process, and how to filter for encounters.

F.1 ADDING LOCAL SENSOR TRACKS TO GLOBAL TRACKS

In order to facilitate fusing of tracks from multiple sensors into global tracks, we break the continental United States into 20 NM by 20 NM bins. Every track is associated to a geographic bin. Whenever the fusion process receives a new local sensor track, or a later report for an as yet unfused local track, an attempt is made to fuse this track to an existing global track. This process is performed by comparing the new track to all neighboring global tracks. The neighboring global tracks for a local track include all global tracks in the same geographical bin as the local track plus global tracks in the surrounding bins (totalling 9 bins). Several tests must be passed for the successful fusion of the single track to a global track:

1. The global track must not already be fused to another local track from the new track's sensor.
2. The tracks must agree on Mode A code (primary-only tracks automatically pass).
3. If the code agreement was on a discrete code, a very coarse horizontal positional test must be passed.
4. If the code agreement was on 1200 code, or no codes, a tighter positional test must be passed, as well as altitude and velocity tests. However, if only the velocity test fails, a potential fusion is declared; 3 successive potential fusions with the same global track results in a successful fusion.

If more than one possible global track satisfies the fusion tests, the one with the highest matching score is chosen. Existing fusion matches are checked each time a new report is received. If the tests fail for three scans in a row, the fusion is ended, and a new global track is sought for the local sensor track.

F.1.1 Code Matching Test

Normally, the code of the new track is the same as the code of the global track. However, code mismatching can complicate the fusion process. Code declaration errors due to data corruption, missing codes, and code changes due to controller action are all common. For this reason, associated with each global track is an established code and an alternate code, which is the code of the most recent report. Usually these two codes are the same. When an alternate code is different from the established code for three successive reports, however, the established code is updated to the alternative code. Reports with no beacon code are ignored in this process. However, if a track has never had been associated with a beacon code report, we consider this track to be a primary-only and give the track an established code of 0.

If both the local track the global track have a beacon code, then a successful code match is declared if any of the following statements are true:

1. The established codes match.
2. The alternate codes match.
3. One of the established codes and the other alternate code match.

However, failure is declared if the match is on code 0, and both tracks have a beacon code in the other code slot that do not match. Lastly, if one track is radar only, and the other track has a beacon code, failure is declared.

To handle local track code changes due to a track swap in the single sensor tracker, confirmation of the code change by the global track is required. If at the time of the local track code change, the global track has had an update by a different sensor's track with the old beacon code, a track swap is declared, and the local track is removed from fusion with the global track. The local track then undergoes a new fusion process.

F.1.2 Horizontal Position Matching Test

Horizontal positional matching requires agreement between the global track's most recent horizontal position x_g, y_g and the new local track's horizontal position x_l, y_l projected back to the time of the global track. This test is simple if both track's reports contained altitude. If a radar report contains altitude h, range ρ, azimuth θ, then the track's horizontal position x, y can be empirically determined.

If, however, the altitude of a track is unknown, then the tracks' altitude has to be assumed (or guessed) in order to derive the track's horizontal position. Simply guessing an altitude may produce a erroneous x, y position. In order to use a reasonable altitude value, we employ the following algorithm:

1. If only one track has known altitude, then we convert the other track's stored ρ, θ position to x, y using the first track's altitude.
2. If neither track has known altitude (which is always true for a primary-only match), we consider all altitudes from 0 NM to 7 NM at 1 NM steps. We then use the altitude that produces the closest positional match between the two tracks. While a smaller step size may produce more accurate estimates, we found that 1 NM is sufficient for fusing two tracks.

Horizontal positional agreement is declared if the horizontal distance between the two tracks is less than an acceptable value:

$$\sqrt{(x_{\mathrm{g}} - x_{\mathrm{l}})^2 + (y_{\mathrm{g}} - y_{\mathrm{l}})^2} \leq \Delta r_{\max} + 3\sigma_{\mathrm{az}}\rho_g + 3\sigma_{\mathrm{az}}\rho_l \,, \tag{F-1}$$

where we use $\Delta r_{\max} = 20\,\mathrm{NM}$ for a discrete code match and $\Delta r_{\max} = 1\,\mathrm{NM}$ for a 1200 code or radar match. The standard deviation for horizontal position error terms σ_{az} account for positional errors due to azimuth noise in radar measurements (which is the dominant source of horizontal position error). We model the standard deviation of the azimuth noise as 3 milliradians for the data format of our radar feed.

F.1.3 Altitude Matching Test

Altitude matching requires agreement between the local and global track altitudes when both are known. Two comparisons are tested; the success of either test results in a match. The comparison test is:

$$\begin{aligned} \Delta t_{\mathrm{l}} &= t_{\mathrm{g}} - t_{\mathrm{l}} \\ \Delta h_{\mathrm{l}} &= |h_{\mathrm{g}} - h_{\mathrm{l}}| \\ \Delta h_{\mathrm{l}} &\leq \Delta h_{\max} \\ \frac{\Delta h_{\mathrm{l}}}{\Delta t_{\mathrm{l}}} &\leq \Delta \dot{h}_{\max} \end{aligned}$$

where we use $\Delta h_{\max} = 600\,\mathrm{ft}$ and $\Delta \dot{h}_{\max} = 100\,\mathrm{ft/s}$. Since the altitude of a track can significantly change between sequential reports, we test both the most recent and previous local track altitudes with the most recent global track altitude update. Only one of the local track altitudes is required to pass the test.

F.1.4 Velocity Matching Test

Velocity matching requires agreement between the two track headings ψ and speeds s according to the following tests:

$$\begin{aligned} |\psi_{\mathrm{g}} - \psi_{\mathrm{l}}| &\leq \Delta \psi_{\max} \\ |s_{\mathrm{g}} - s_{\mathrm{l}}| &\leq \Delta s_{\max} \\ \frac{1}{2} \leq \tfrac{s_{\mathrm{g}}}{s_{\mathrm{l}}} &\leq 2 \end{aligned}$$

where we use $\Delta\psi_{\max} = 45°$ and $\Delta s_{\max} = 100\,\mathrm{kt}$. The last test is needed for slow aircraft and clutter tracks, to prevent, for example, speeds of 20 and 110 kt from agreeing.

F.2 DETERMINING TRACK AIRSPEED AND HEADING

Determining a global track's airspeed and heading is a two step process. First, the individual sensor tracks are smoothed. Second, the individual track are averaged using relative weights that account for sensor update times and the quality of each sensor's measurement. We apply both alpha smoothing and curve fitting to determine airspeed and heading, depending upon the track situation. We have implemented various maneuver detection algorithms, and tracking is dependent upon the current turn rate and acceleration states of the track.

F.2.1 Local Track Smoothing

First, we require that the track has moved a minimum distance for it to be considered. If the track never moves more than 1 NM, then the track is thrown out. After the movement test is satisfied, the track's airspeed and heading are calculated from the new and previous positions. We then update the local track's airspeed and heading estimates using alpha smoothing.

First, the current heading estimate $\psi^{(j)}$, and its difference from the previous estimate $\psi^{(j-1)}$, are given by

$$\begin{aligned}\psi^{(j)} &= \operatorname{atan2}\Big((x^{(j)} - x^{(j-1)}), (y^{(j)} - y^{(j-1)})\Big)\\ \Delta\psi^{(j)} &= \psi^{(j)} - \psi^{(j-1)}\end{aligned}$$

Next, we determine the current turn rate state S_ψ of the track:

$$S_\psi^{(n)} = \begin{cases} 2 & \text{if } \Delta\psi^{(j)} > \sigma_{\text{heading}} \\ 1 & \text{if } \Delta\psi^{(j)} > \Delta\psi_{\min} \\ -2 & \text{if } \Delta\psi^{(j)} < -\sigma_{\text{heading}} \\ -1 & \text{if } \Delta\psi^{(j)} < -\Delta\psi_{\min} \\ 0 & \text{otherwise} \end{cases} \qquad \text{(F-2)}$$

where we use $\Delta\psi_{\min} = 3^\circ$ and σ_{heading} is the standard deviation of the heading noise, which is calculated from the standard deviations for range and azimuth noise of the sensors. Note that a positive $\Delta\psi$ value corresponds to a right turn, while a negative value corresponds to a left turn. We then use $S_\psi^{(j)}$ to determine the smoothing value alpha α in Table F-1 of the individual tracks that will be used to calculate the heading of the global track at the current time.

The new track heading is finally given by:

$$\psi^{(j)} = \psi^{(j-1)} + \alpha \times \Delta\psi^{(j)}$$

and we iterate through this process for the entire track.

The process to estimate airspeed s is similar, with one important difference. If successive positions are simply connected, then the airspeed estimates will always be too high, since the aircraft will appear to "zig-zag" along the track. Thus, only the projection of the velocity vector

TABLE F-1

Smoothing values depending on the current and previous turn states.

	Current Turn State				
Previous State	Large Left Turn (-2)	Small Left Turn (-1)	No Turn (0)	Small Right Turn (+1)	Large Right Turn (+2)
Large Left (-2)	0.7	0.7	0.4	0.5	0.5
Small Left (-1)	0.7	0.4	0.4	0.5	0.5
No Turn (0)	0.4	0.4	0.3	0.4	0.4
Small Right (+1)	0.5	0.5	0.4	0.4	0.7
Large Right (+2)	0.5	0.5	0.4	0.7	0.7

onto the track's heading vector is used to determine the track's airspeed:

$$\begin{aligned} s^{(j)} &= \cos\left(\frac{\Delta\psi^{(j)}}{2}\right) \times \sqrt{\frac{(x^{(j)} - x^{(j-1)})^2 + (y^{(j)} - y^{(j-1)})^2}{t^{(j)} - t^{(j-1)}}} \\ \Delta s^{(j)} &= s^{(j)} - s^{(j-1)} \end{aligned}$$

We then determine the current airspeed acceleration state S_s of the track using a similar technique as we did for turn rate.

$$S_s^{(j)} = \begin{cases} 2 & \text{if } \Delta s^{(j)} > \sigma_{\text{speed}} \\ 1 & \text{if } \Delta s^{(j)} > \Delta s_{\text{min}} \\ -2 & \text{if } \Delta s^{(j)} < -\sigma_{\text{heading}} \\ -1 & \text{if } \Delta s^{(j)} < -\Delta s_{\text{min}} \\ 0 & \text{otherwise} \end{cases}$$

where we use $\Delta s_{\text{min}} = 18\,\text{kt}$ and σ_{speed} is the standard deviation of airspeed error due to noise in range and azimuth measurements from the radar sensors. The speed smoothing and the speed alpha table rules are the same as for the heading case.

F.2.2 Global Track Smoothing

In order to determine a global track's airspeed and heading at each measurement, we use a weighted least squares estimation approach. In this section we describe in detail the approach for determining the tracks heading.

First, each sensor's heading estimate is assigned a weight w_i at the current time $t^{(c)}$ as follows:

$$w_i^{(c)} = \sigma_{\text{heading}}^{-1} \times \frac{t_{\text{max}} - \frac{t_i^{(j)} + t_i^{(j-1)}}{2} - t^{(c)}}{t_{\text{max}}}$$

where σ_{heading} is the standard deviation of the heading noise and $t_{\max} = 18\,\text{s}$ is a discounting factor that takes into account the time difference between the measurement from the sensor being considered and the time for when we are determining heading. The time $t_i^{(j)}$ corresponds to the time of the next closest measurement for sensor i with respect to the current time that we are trying to determine the track's heading.

Next, we determine the total turn state score for the track

$$\left| \sum_{i=1}^{N} S_i \right| ,$$

where N is the number of sensors supporting the track and S_i is the current turn state value for the ith sensor defined in Equation F-2. If the turn rate score is less than N, then we consider the track to be non-turning and the current global heading is simply the weighted average of the N sensor heading estimates:

$$\psi_{\text{global}}^{(c)} = \frac{\sum_{i=1}^{N} \psi_i^{(j)} \times w_i^{(j)}}{\sum_{i=1}^{N} w_i^{(c)}}$$

Otherwise, if the turn rate score is greater than or equal to N, then we consider the track to be in a turn. In this case, we utilize weighted least squares estimated to determine a first-order relationship between time and heading.

The global track speed calculation is identical in form to the global track heading calculation.

This page intentionally left blank.

APPENDIX G
SENSORS

The following table lists the RADES sensors that contributed to the model. Shown are the number of hours of flight time by transponder-equipped aircraft squawking 1200 captured by each sensor between 1 December 2007 and 7 December 2007 and between 1 June 2008 and 7 June 2008. There was a total of 200,000 raw 1200-code hours from 130 sensors. Note that after fusion, these 200,000 hours of reports correspond to only 74,000 flight hours.

TABLE G-1

Beacon hours.

60 NM range sensors	
1131.05	ACT Waco, TX, USA (ASR-11)
284.29	ADW Camp Springs (Andrews AFB), MD, USA (ASR-9)
26.23	BGR Bangor, ME, USA (ASR-8)
965.78	BOS Logan Intl (Boston), MA, USA (ASR-9)
575.35	BUF Buffalo Intl, NY, USA (ASR-9)
1055.71	BWI Baltimore (BWI), MD, USA (ASR-9)
782.69	COS Colorado Springs, CO, USA (ASR-11)
477.42	COU Columbia MO, MO, USA (ASR-11)
213.16	CRW Charleston, WV, USA (ASR-8)
254.29	DCA Washington National, DC, USA (ASR-9)
1006.08	DOV Dover AFB, DE, USA (GPN-20)
2116.03	EWR Newark, NJ, USA (ASR-9)
1785.01	FLL Ft Lauderdale (Hollywood Intl), FL, USA (ASR-9)
912.35	GRK Ft Hood, TX, USA (ASR-9)
2494.87	HPN White Plains (Westchester Co), NY, USA (ASR-9)
472.2	HUF Terre Haute, IN, USA (ASR-8)
802.05	IAD Washington Dulles (Chantilly), DC, USA (ASR-9)
2599.42	JFK New York (JFK), NY, USA (ASR-9)
4318.76	LAXB Los Angeles Intl - North, CA, USA (ASR-9)
506.15	LFT Lafayette, LA, USA (ASR-11)
1519.64	MDT Harrisburg, PA, USA (ASR-9)
1607.94	MHT Manchester, NH, USA (ASR-9)
310.94	MLU Monroe LA, LA, USA (ASR-8)
707.38	MRB Martinsburg, WV, USA (ASR-9)
203.04	MUO Mountain Home AFB, ID, USA (GPN-20)
137.65	NQXA Key West NAS, FL, USA (GPN-27)

Continued on next page...

TABLE G-1. Continued

431.77	PWM Portland (Cumberland), ME, USA (ASR-9)
73.94	RCA Ellsworth AFB (Rapid City), SD, USA (GPN-20)
457.75	SAV Savannah, GA, USA (ASR-8)
71.62	SAW Marquette (KI Sawyer), MI, USA (ASR-7F)
2419.56	SCK Stockton, CA, USA (ASR-11)
529.39	SDF Louisville, KY, USA (ASR-9)
374.99	SJU San Juan, PR, USA (ASR-8)
446.86	STT St.Thomas, PR, USA (ASR-8)
1164.49	SWF Newburgh (Stewart), NY, USA (ASR-9)
3994.25	XMR Cape Canaveral, FL, USA (GPN-30)
	120 NM range sensors
810.42	NHK Patuxent River NAS, MD, USA (ASR-11)
	200 NM range sensors
547.92	AEX Alexandria, LA, USA (AN/FPS-20A)
208.73	AMA Amarillo, TX, USA (AN/FPS-67B)
4793.99	ATL Atlanta (Marietta), GA, USA (ARSR-1E)
349.35	BAM Battle Mountain, NV, USA (ARSR-2)
977.4	CDC Cedar City, UT, USA (ARSR-2)
828.06	CPV Coopersville, MI, USA (AN/FPS-66A)
554.87	DSV Buffalo (Dansville), NY, USA (ARSR-1E)
347.89	FLX Fallon, NV, USA (AN/FPS-66A)
1602.96	FPK Salt Lake City (Francis Peak), UT, USA (ARSR-1E)
4930.86	FTW Ft Worth (Keller), TX, USA (ARSR-1E)
990.45	GUP Gallup (Farmington), NM, USA (ARSR-2)
2693.26	HOU Houston (Ellington), TX, USA (ARSR-1E)
1277.87	IND Indianapolis, IN, USA (ARSR-1E)
469.98	IRK Kirksville, MO, USA (ARSR-3)
2250.79	JOL Elwood (Joliet), IL, USA (ARSR-3)
376.86	LSK Lusk, WY, USA (ARSR-2)
1512.66	MGM Montgomery, AL, USA (ARSR-1E)
1479.68	PIT Pittsburgh (Oakdale), PA, USA (AN/FPS-67B)
3697.68	QAS Las Vegas (Angel Peak), NV, USA (AN/FPS-20E)
3266.18	QBE Roanoke (Bedford), VA, USA (ARSR-3)
2184.12	QBN Binns Hall, VA, USA (ARSR-3)
1471.06	QBZ Oskaloosa, KS, USA (ARSR-2)
1011.95	QCF Du Bois (Clearfield), PA, USA (ARSR-3)
691.05	QCK Boise (Cascade), ID, USA (ARSR-2)
1999.77	QDB Cleveland (Brecksville), OH, USA (ARSR-1E)

Continued on next page...

TABLE G-1. Continued

1636.85	QDT Detroit (Canton), MI, USA (ARSR-1E)
2473.14	QHA Cummington, MA, USA (AN/FPS-67B)
636.73	QHB St. Albans, VT, USA (AN/FPS-67B)
1662.71	QHZ Horicon, WI, USA (ARSR-2)
197.26	QJB Gettysburg, SD, USA (AN/FPS-67B)
1562.55	QJE Minneapolis (Apple Valley), MN, USA (ARSR-1E)
632.06	QJO Arlington, IA, USA (ARSR-3)
695.32	QJQ San Juan (Pico del Este), PR, USA (AN/FPS-20E)
1968.31	QNK Lincolnton, GA, USA (ARSR-3)
1026.86	QNM Newport, MS, USA (ARSR-3)
1911.52	QOJ Nashville (Joelton), TN, USA (ARSR-1E)
1767.98	QPC Haleyville, AL, USA (AN/FPS-67B)
2953.44	QPK Denver (Parker), CO, USA (ARSR-1E)
3116.41	QPL The Plains, VA, USA (ARSR-3)
2877.7	QRB Citronelle, AL, USA (ARSR-2)
1316.35	QRC Benton, PA, USA (AN/FPS-67B)
1753.04	QRI Lynch, KY, USA (ARSR-2)
1991.76	QRL Raleigh (Benson), NC, USA (ARSR-1E)
1302.22	QRM Charlotte (Maiden), NC, USA (ARSR-1E)
892.99	QSA Albuquerque (West Mesa), NM, USA (AN/FPS-66A)
328.34	QSI Lovell, WY, USA (ARSR-2)
881.87	QSR Boron, CA, USA (AN/FPS-67B)
1399.72	QTZ La Grange, IN, USA (ARSR-1E)
811.06	QUZ Hanna City, IL, USA (AN/FPS-67B)
425.52	QWC Mesa Rica, NM, USA (ARSR-1E)
1830.23	QWO London, OH, USA (ARSR-1E)
935.7	QXR Russellville, AR, USA (AN/FPS-64A)
716.15	QXS Odessa (Andrews), TX, USA (ARSR-1E)
872.87	QYB Byhalia (Memphis), MS, USA (ARSR-1E)
2727.11	QYS Rogers, TX, USA (ARSR-1E)
1794.1	SEA Seattle (Ft Lawton), WA, USA (ARSR-1E)
1781.55	STL St. Louis, MO, USA (ARSR-1E)
631.17	SVC Silver City, NM, USA (ARSR-2)
	250 NM range sensors
3210.22	AJO Ajo, AZ, USA (ARSR-4)
21.57	BAR Barrington, NS, Canada (AN/FPS-117)
2243.34	CTY Cross City, FL, USA (ARSR-4)
735.48	DMN Deming (Magdelina Peak), NM, USA (ARSR-4)
624.6	LCH Lake Charles, LA, USA (ARSR-4)

Continued on next page. . .

TABLE G-1. Continued

7521.59	MLB Melbourne, FL, USA (ARSR-4)
3573.05	NEN Jacksonville (White House Fl), FL, USA (ARSR-4)
1212.88	NEW New Orleans (Slidell), LA, USA (ARSR-4)
789.25	NQX Key West, FL, USA (ARSR-4)
5730.13	NSD San Clemente Island, CA, USA (ARSR-4)
1335.16	PAM Panama City (Tyndall), FL, USA (ARSR-4)
2492.48	PRB Paso Robles, CA, USA (ARSR-4)
1844.45	QEA North Truro, MA, USA (ARSR-4)
7730.73	QFN Ft Green, FL, USA (ARSR-4)
4168.08	QIE Gibbsboro, NJ, USA (ARSR-4)
588.16	QJA Empire, MI, USA (ARSR-4)
363.15	QJD Nashwauk, MN, USA (ARSR-4)
337.08	QKW Makah, WA, USA (ARSR-4)
6114.05	QM8 Tamiami, FL, USA (ARSR-4)
7649.78	QMV Mill Valley, CA, USA (ARSR-4)
1386.96	QNA Morales, TX, USA (ARSR-4)
459.49	QNW Eagle Peak, TX, USA (ARSR-4)
980.37	QOM King Mountain, TX, USA (ARSR-4)
855.56	QRJ Charleston (Jedburg), SC, USA (ARSR-4)
3186.88	QRW Mt. Laguna, CA, USA (ARSR-4)
3755.59	QVH Riverhead, NY, USA (ARSR-4)
1347.25	QVR Oceana, VA, USA (ARSR-4)
99.11	QWA Watford City, ND, USA (ARSR-4)
339.11	QXU Utica (Remsen), NY, USA (ARSR-4)
91.52	QYA Bucks Harbor, ME, USA (ARSR-4)
69.39	QYD Caribou, ME, USA (ARSR-4)
436.17	QZA Oilton, TX, USA (ARSR-4)
298.93	QZZ Rainbow Ridge, CA, USA (ARSR-4)
415.94	RSG Rocksprings, TX, USA (ARSR-4)
1428.13	SLE Salem (Laurel Mountain), OR, USA (ARSR-4)

APPENDIX H
DENSITY PLOTS

The figures in this appendix show the density of cooperative aircraft squawking 1200 (VFR) in the radar data used to construct the model. Similar density data for noncooperative aircraft are also available but require significantly more processing due to clutter. Density is measured in average number of aircraft per NM^3. Each pixel measures 1/6 of a degree on a side in latitude and in longitude. Depending on latitude, the square mileage of a pixel may be between 64 NM^2 at the northernmost latitude to 92 NM^2 at the southernmost latitude. The highest density pixel is in Miami-Dade County and has a density of approximately 0.003 aircraft per NM^3 over all altitude layers between 500 ft AGL and FL180, with the highest density layer between 500 ft AGL and 1200 ft AGL.

It is important to emphasize that this density map is the observed density from the radar data set. Factors such as terrain and curvature of the earth prevent the observation of some aircraft, particularly low flying aircraft such as those flying VFR. The locations of the sensors are indicated on the maps by magenta circles. It is no coincidence that the measured density in the lowest altitude layer, for example, tend to be concentrated around the locations of the sensors. In particular, coverage at low altitudes tends to be sparser in the western portion of the country. Coverage above 5000 ft AGL, however, is fairly good across the country.

Certain areas on the maps are worth attention. Due to the protected area around Washington DC, there are virtually no 1200 squawking aircraft at any altitude layer in that region. By contrast, there are is a large concentration in Florida and southern Arizona, especially near the two locations of the Embry-Riddle flight school in Daytona Beach, Florida and Prescott, Arizona. In addition, the effect of the "upside down wedding-cake" airspace class structure around major airports can be seen if one compares the heavy concentration near the pixel representing Atlanta and Dallas at lower altitude layers against the empty airspace in the same pixel at the highest altitude layer. Finally, it can be seen that the density near Denver at higher altitudes AGL is quite low; this may be attributed to the fact that an additional 5,000 ft above ground in that region puts the aircraft above 10,000 ft MSL.

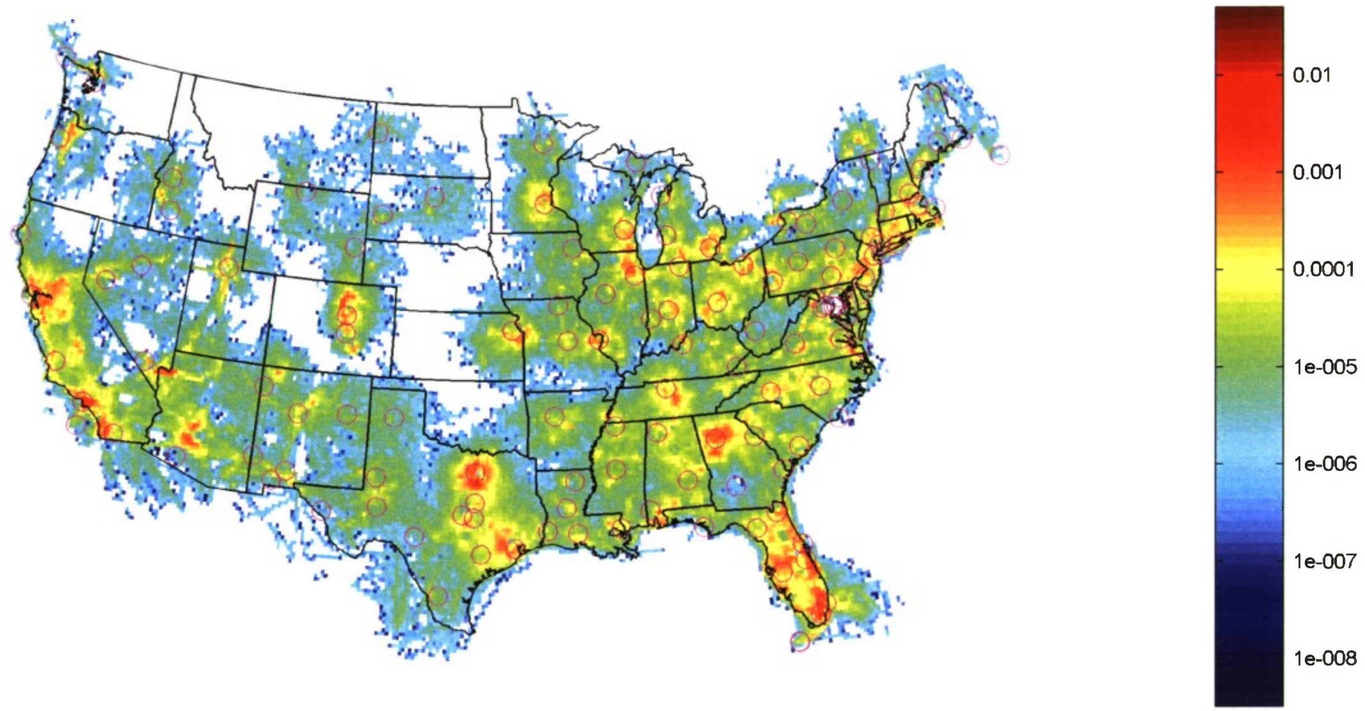


Figure H-1. Average 1200-code cooperative density over all altitude layers up to FL180 in aircraft per NM^3 for the periods 1–7 December 2007 and 1–7 June 2008. White indicates no VFR traffic reports in that pixel.

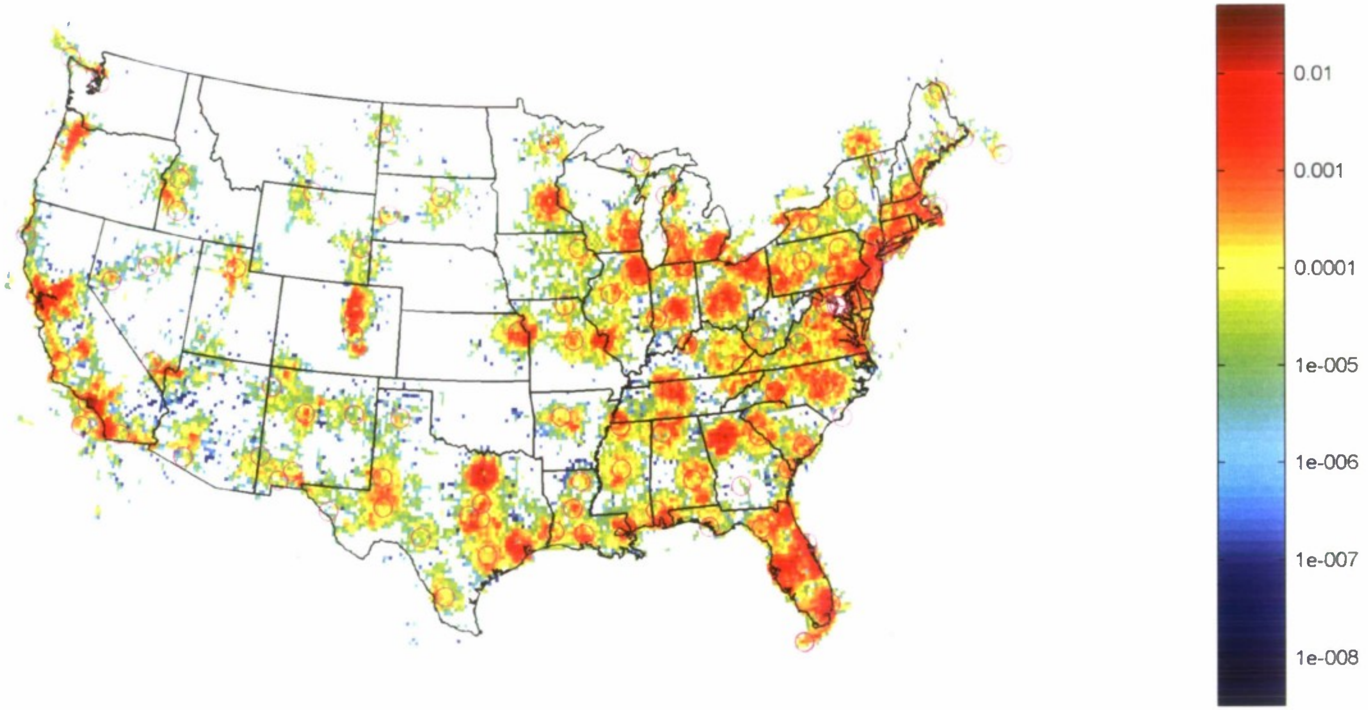


Figure H-2. Average 1200-code cooperative density between 500 and 1200 ft AGL in aircraft per NM^3 for the periods 1–7 December 2007 and 1–7 June 2008. White indicates no VFR traffic reports in that pixel.

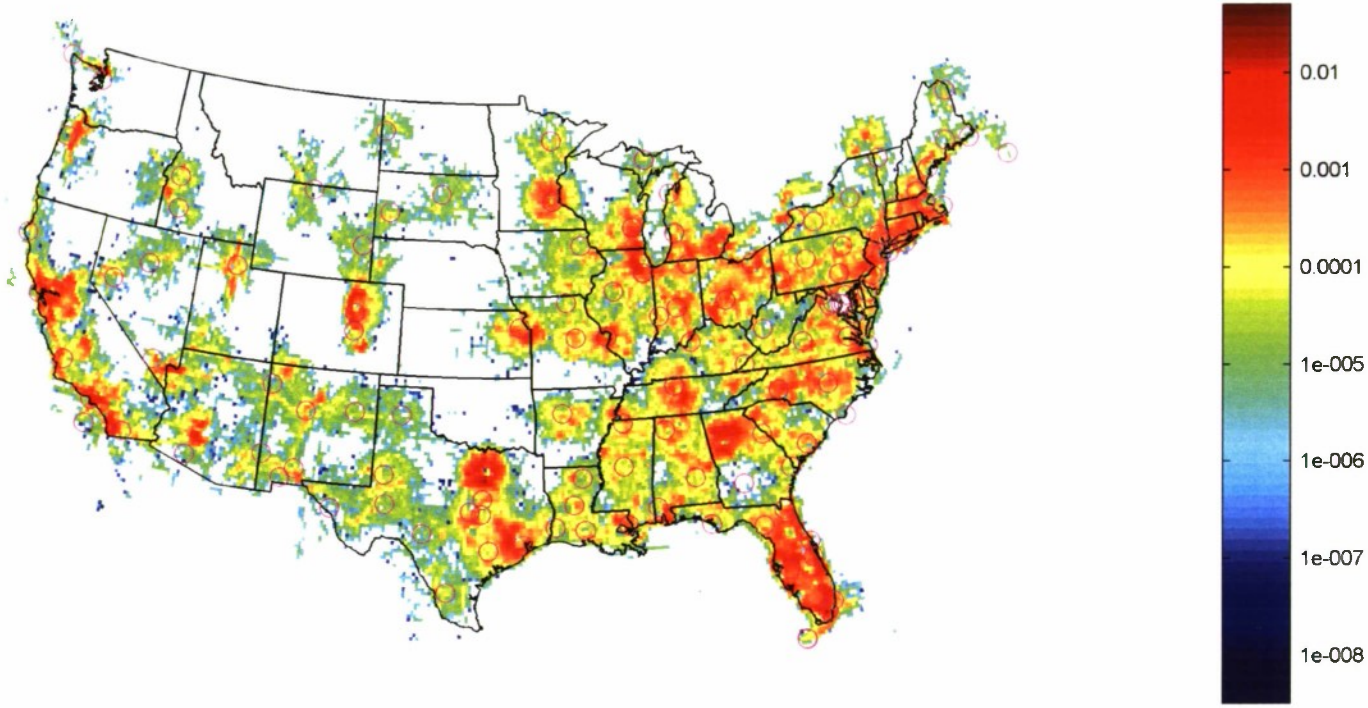


Figure H-3. Average 1200-code cooperative density between 1200 and 3000 ft AGL in aircraft per NM^3 for the periods 1–7 December 2007 and 1–7 June 2008. White indicates no VFR traffic reports in that pixel.

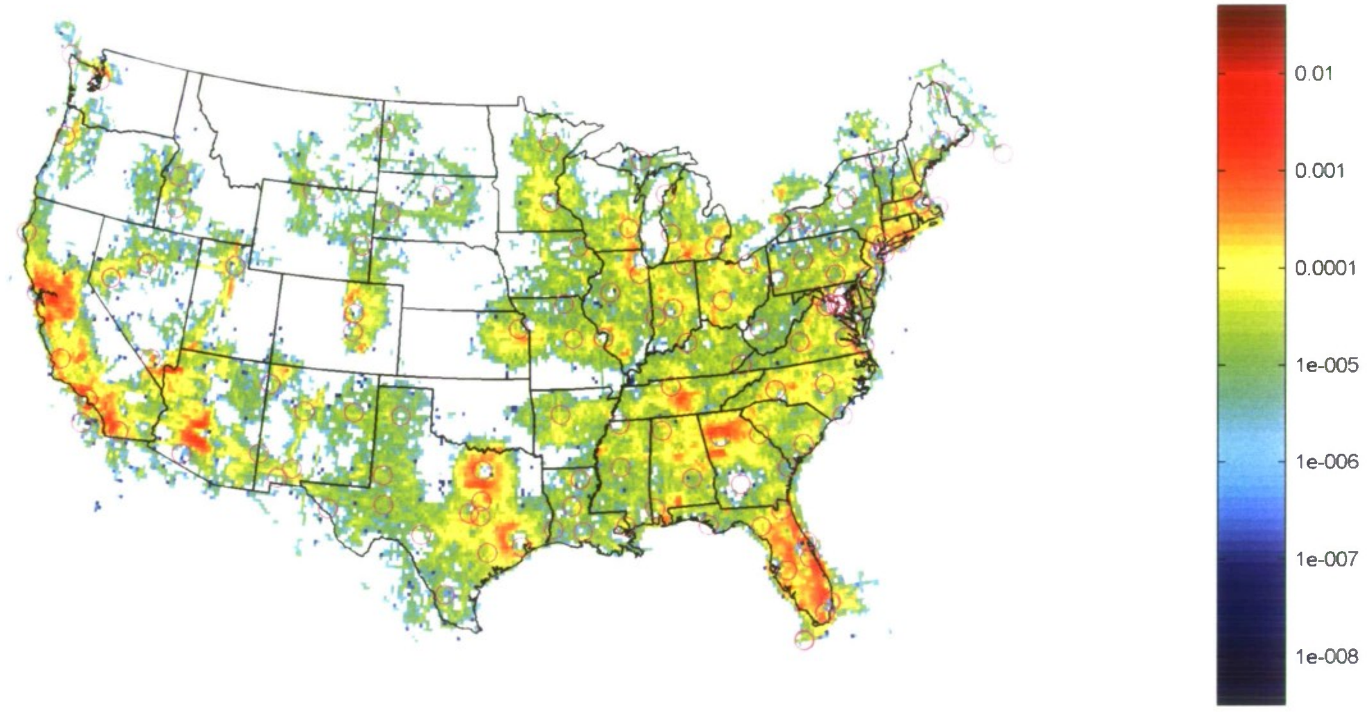


Figure H-4. Average 1200-code cooperative density between 3000 and 5000 ft AGL in aircraft per NM^3 for the periods 1–7 December 2007 and 1–7 June 2008. White indicates no VFR traffic reports in that pixel.

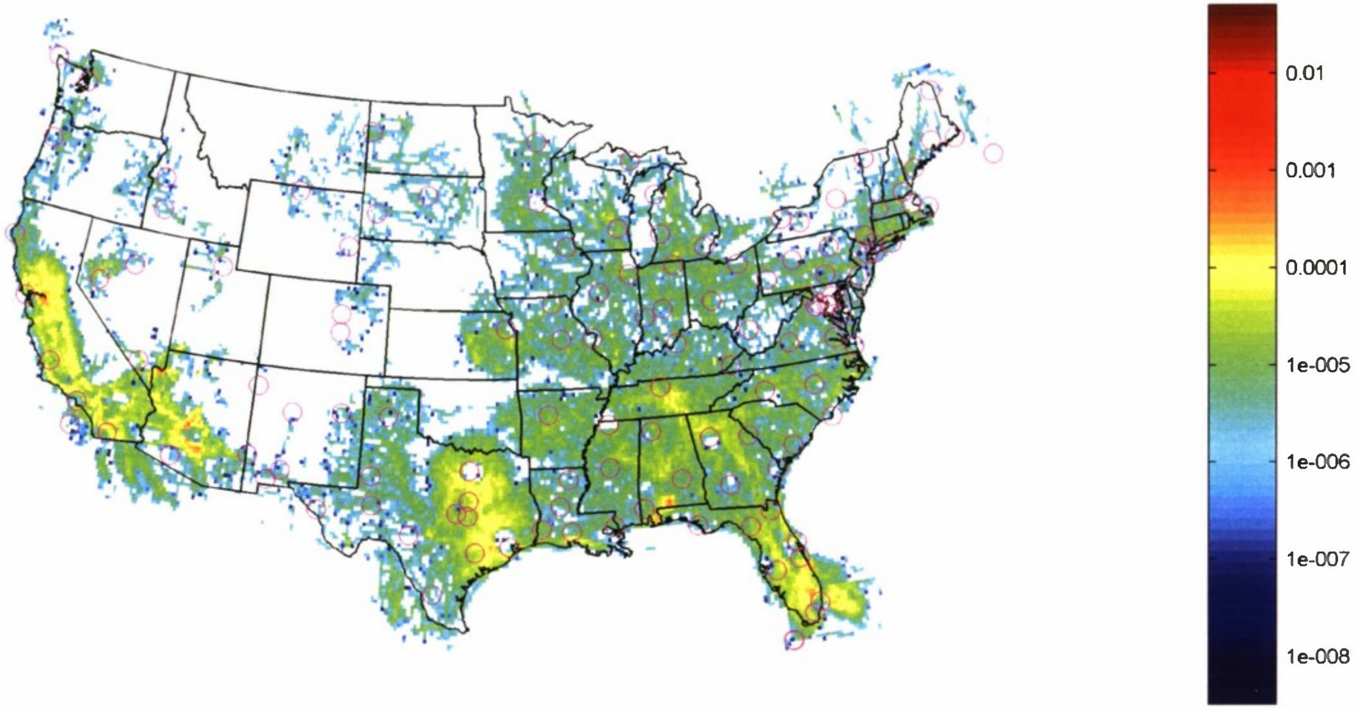


Figure H-5. Average 1200-code cooperative density between 5000 ft AGL and 10,000 ft MSL in aircraft per NM^3 for the periods 1–7 December 2007 and 1–7 June 2008. White indicates no VFR traffic reports in that pixel.

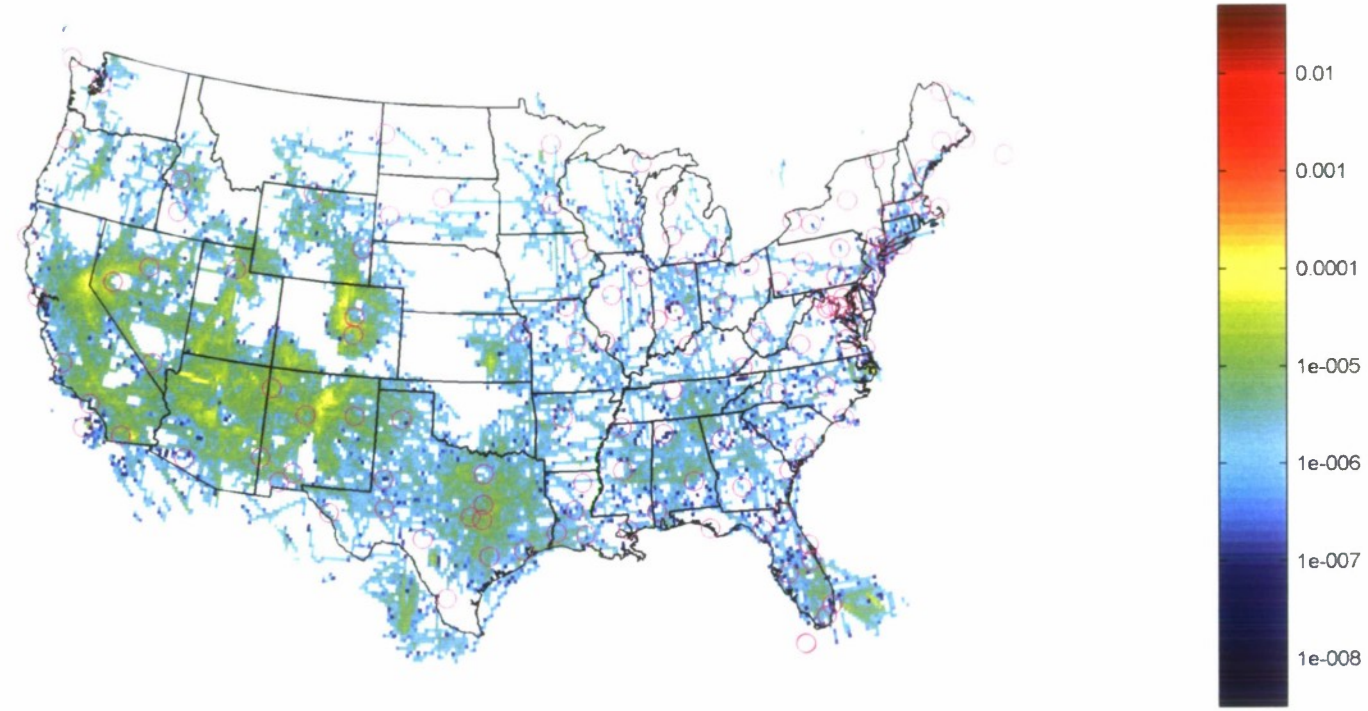


Figure H-6. Average 1200-code cooperative density between 10,000 ft MSL and FL180 in aircraft per NM^3 for the periods 1–7 December 2007 and 1–7 June 2008. White indicates no VFR traffic reports in that pixel.

This page intentionally left blank.

APPENDIX I
BAYESIAN NETWORKS

This appendix briefly reviews Bayesian networks. Further discussion of Bayesian networks may be found elsewhere [23–25].

I.1 DEFINITION

A Bayesian network is a graphical representation of a multivariate probability distribution over variables $\boldsymbol{X} = X_1, \ldots, X_n$. In particular, a Bayesian network is a directed acyclic graph G whose nodes correspond to variables and edges correspond to probabilistic dependencies between them. Associated with each variable X_i is a conditional probability distribution $P(x_i \mid \boldsymbol{\pi}_i)$, where $\boldsymbol{\pi}_i$ denotes an instantiation of the parents of X_i in the graph. The probability of an instantiation of the variables is specified directly by the conditional probability distributions in the Bayesian network:

$$P(\boldsymbol{x}) = P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i \mid \boldsymbol{\pi}_i)\,. \tag{I-1}$$

I.2 SAMPLING

It is rather straightforward to sample from a multivariate distribution represented by a Bayesian network. The first step is to produce a topological sort of the nodes in the network. A topological sort orders the nodes in a Bayesian network such that if a node X_i comes before X_j there does not exist a directed path from X_j to X_i. Every Bayesian network has at least one topological sort, but there may be many. Efficient algorithms exist for finding a valid topological sort [26].

To produce a sample from the joint distribution represented by a Bayesian network, we simply iterate through a topologically sorted sequence of the variables and sample from their conditional probability distributions. The topological sort ensures that when sampling from each conditional probability distribution the necessary parents have been instantiated.

I.3 PARAMETER LEARNING

The parameters $\boldsymbol{\theta}$ of a Bayesian network determine the associated conditional probability distributions. Given some fixed network structure G, we can learn these parameters from data. In this appendix, we assume that the variables are discrete.

Before discussing how to learn the parameters of a Bayesian network, it is necessary to introduce some notation. Let r_i represent the number of instantiations of X_i and q_i represent the number of instantiations of the parents of X_i. If X_i has no parents, then $q_i = 1$. The jth instantiation of the parents of X_i is denoted $\boldsymbol{\pi}_{ij}$.

There are $\sum_{i=1}^{n} r_i q_i$ parameters in a Bayesian network. Each parameter is written θ_{ijk} and determines $P(X_i = k \mid \boldsymbol{\pi}_{ij})$, i.e.,

$$P(X_i = k \mid \boldsymbol{\pi}_{ij}) = \theta_{ijk} \,.$$

Although there are $\sum_{i=1}^{n} r_i q_i$ parameters, only $\sum_{i=1}^{n} (r_i - 1) q_i$ are independent.

Computing the posterior $p(\boldsymbol{\theta} \mid D, G)$ involves specifying a prior $p(\boldsymbol{\theta} \mid G)$ and applying Bayes' rule

$$p(\boldsymbol{\theta} \mid D, G) = \frac{P(D \mid \boldsymbol{\theta}, G) p(\boldsymbol{\theta} \mid G)}{P(D \mid G)} = \frac{P(D \mid \boldsymbol{\theta}, G) p(\boldsymbol{\theta} \mid G)}{\int P(D \mid \boldsymbol{\theta}, G) p(\boldsymbol{\theta} \mid G) \, \mathrm{d}\boldsymbol{\theta}} \,. \tag{I-2}$$

If N_{ijk} is the count of $X_i = k$ given $\boldsymbol{\pi}_{ij}$ in the data D, then the probability of the data given the parameters $\boldsymbol{\theta}$ is

$$P(D \mid \boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \,. \tag{I-3}$$

Let $\boldsymbol{\theta}_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i})$. Since $\boldsymbol{\theta}_{ij}$ is independent of $\boldsymbol{\theta}_{i'j'}$ when $ij \neq i'j'$, the prior probability of the parameters assuming a fixed structure G is

$$p(\boldsymbol{\theta} \mid G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\boldsymbol{\theta}_{ij} \mid G) \,. \tag{I-4}$$

The density $p(\boldsymbol{\theta}_{ij} \mid G)$ is a distribution over relative frequencies. Under some very weak assumptions, it is possible to prove that $p(\boldsymbol{\theta}_{ij} \mid G)$ is Dirichlet (see [25], Section 6.2.3). Hence,

$$p(\boldsymbol{\theta}_{ij} \mid G) = \begin{cases} \frac{\Gamma(\alpha_{ij0})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} & \text{if } 0 \leq \theta_{ijk} \leq 1 \text{ and } \sum_{k=1}^{r_i} \theta_{ijk} = 1 \\ 0 & \text{otherwise} \end{cases} ,$$

where $\alpha_{ij1}, \ldots, \alpha_{ijr_i}$ are the parameters of the Dirichlet distribution and $\alpha_{ij0} = \sum_{k=1}^{r_i} \alpha_{ijk}$. For the prior to be objective (or noninformative), the parameters α_{ijk} must be identical for all k. Different objective priors have been used in the literature. Cooper and Herskovits [20] use $\alpha_{ijk} = 1$. Heckerman, Geiger, and Chickering [27] use and justify $\alpha_{ijk} = 1/(r_i q_i)$.

It is possible to show that $p(\boldsymbol{\theta}_{ij} \mid D, G)$ is Dirichlet with parameters $\alpha_{ijk} + N_{ijk}, \ldots, \alpha_{ijk} + N_{ijk}$. Hence,

$$p(\boldsymbol{\theta}_{ij} \mid D, G) = \begin{cases} \frac{\Gamma(\alpha_{ij0} + N_{ij})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk} + N_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk} + N_{ijk} - 1} & \text{if } 0 \leq \theta_{ijk} \leq 1 \text{ and } \sum_{k=1}^{r_i} \theta_{ijk} = 1 \\ 0 & \text{otherwise} \end{cases} ,$$

where $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Sampling from a Bayesian network with G known, $\boldsymbol{\theta}$ unknown, and D observed involves assigning k to X_i with probability

$$P(X_i = k \mid \pi_{ij}, D, G) = \int \theta_{ijk} p(\theta_{ijk} \mid D, G) \, \mathrm{d}\theta_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k'=1}^{r_i} (\alpha_{ijk'} + N_{ijk'})} \,. \tag{I-5}$$

I.4 STRUCTURE LEARNING

Finding the most likely structure G that generated a set of data D. The objective is to find the most likely graph given data. By Bayes' rule,

$$P(G \mid D) \propto P(G)P(D \mid G) = P(G) \int P(D \mid \boldsymbol{\theta}, G)p(\boldsymbol{\theta} \mid G)\,\mathrm{d}\boldsymbol{\theta}\,. \tag{I-6}$$

The previous section explains how to compute the likelihood $P(D \mid \boldsymbol{\theta}, G)$ and the prior $p(\boldsymbol{\theta} \mid G)$. Cooper and Herskovits [20] show how to evaluate the integral above, resulting in

$$P(G \mid D) = P(G) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}\,, \tag{I-7}$$

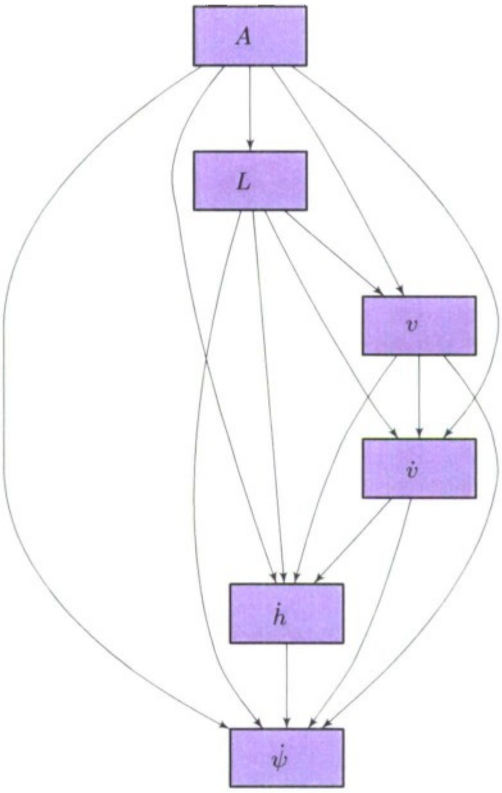
where $N_{ij} = \sum_{k=1}^{r} N_{ijk}$. Heckerman, Geiger, and Chickering [27] suggest priors over graphs, but it is not uncommon in the literature to assume a uniform prior. For numerical convenience, most Bayesian network learning packages calculate and report $\log P(G \mid D) + K$, where K is a constant independent of G. This quantity is often called the *Bayesian score* and may be used for structure comparison and search.

This page intentionally left blank.

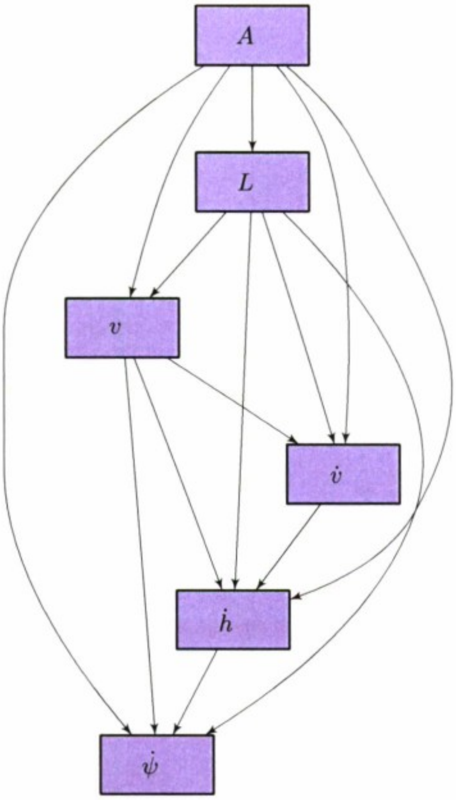
APPENDIX J
NETWORK CANDIDATES

This section displays a selection of various network structures along with their log-Bayesian score, number of edges, and number of parameters. Bayesian model selection balances the complexity of the model according to the amount of data available.

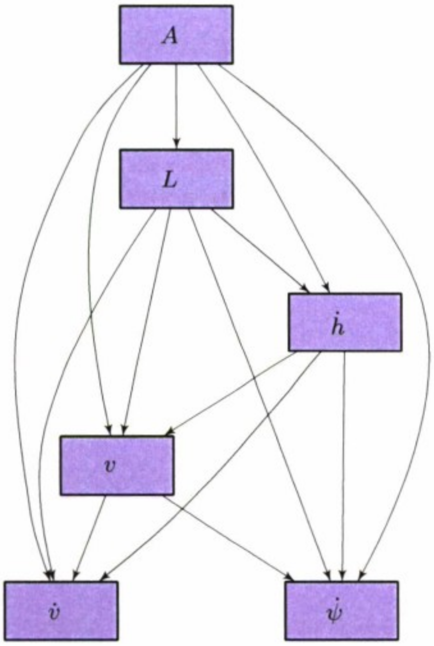
J.1 INITIAL NETWORK CANDIDATES



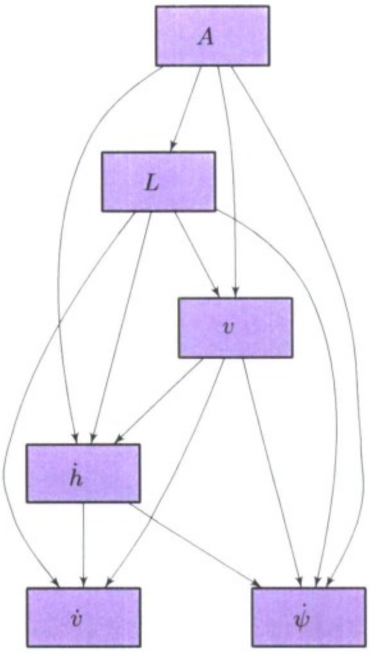
Score = -961638874 (best), Edges = 15, Parameters = 31359



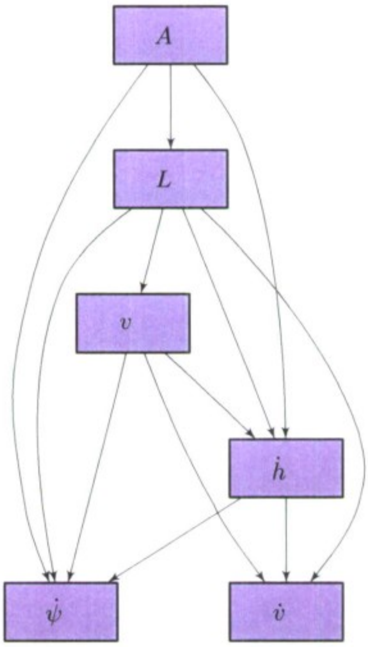
Score = -966691272, Edges = 14, Parameters = 9855



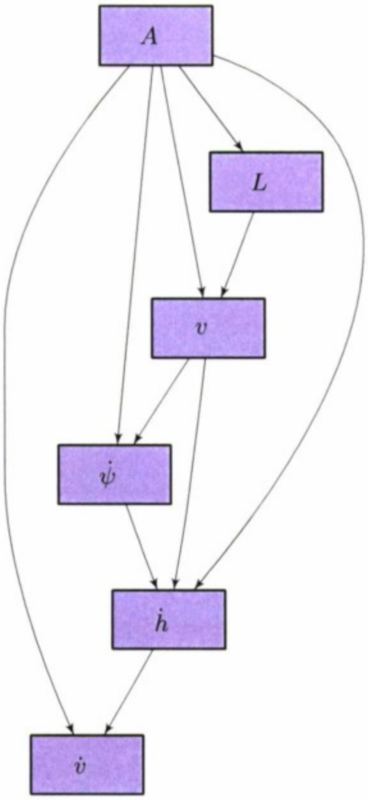
Score = -966694938, Edges = 14, Parameters = 9855



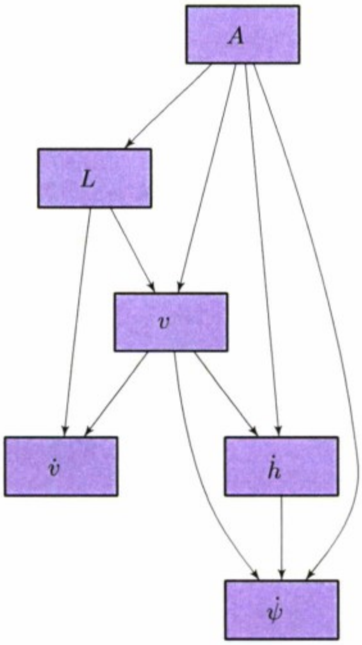
Score = -967537835, Edges = 13, Parameters = 7167



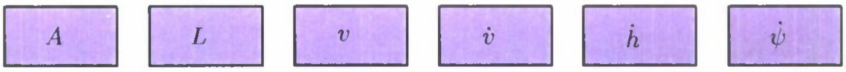
Score = -968020869, Edges = 12, Parameters = 7083



Score = -974539518, Edges = 10, Parameters = 1775

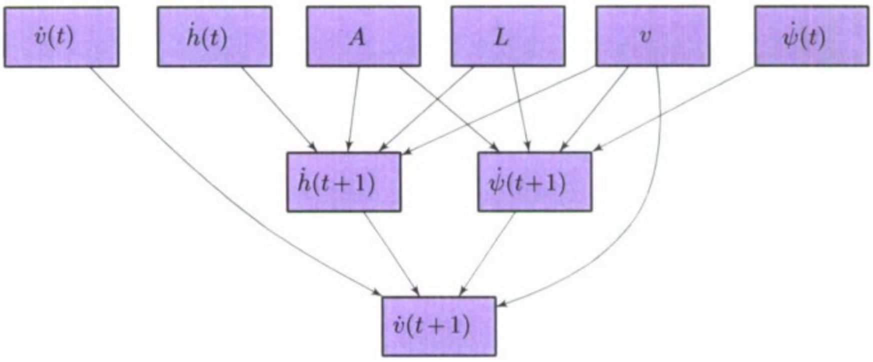


Score = -975871704, Edges = 10, Parameters = 1791

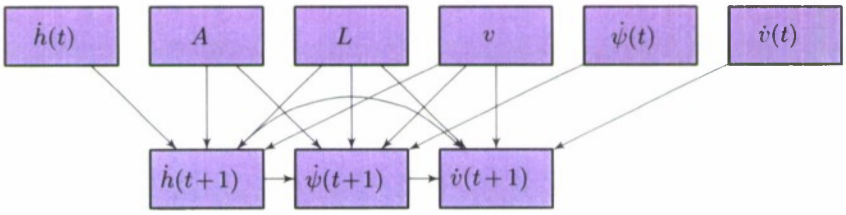


Score = -1012326420, Edges = 0, Parameters = 29

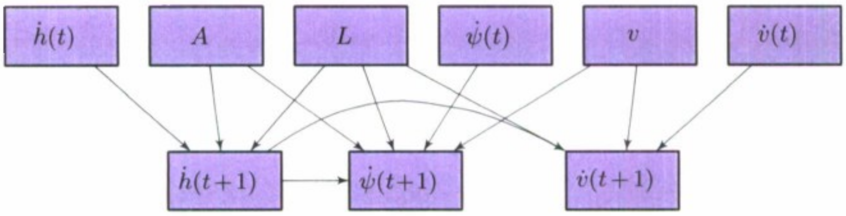
J.2 TRANSITION NETWORK CANDIDATES



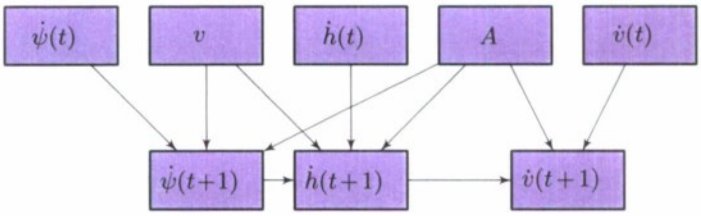
Score = -29174148 (best), Edges = 12, Parameters = 18592



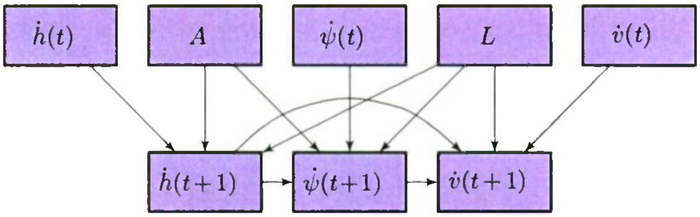
Score = -29214567, Edges = 14, Parameters = 74368



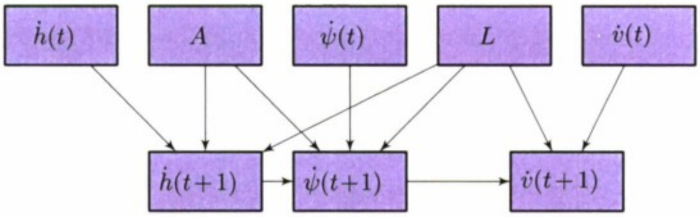
Score = -29252097, Edges = 12, Parameters = 42784



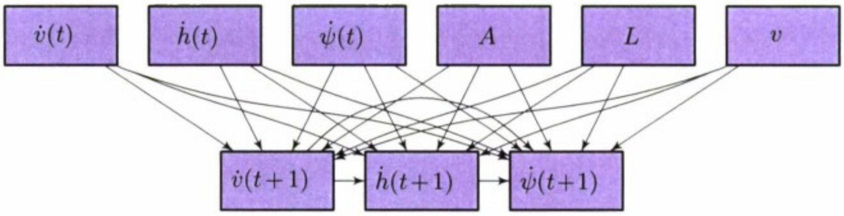
Score = -29391815, Edges = 10, Parameters = 11312



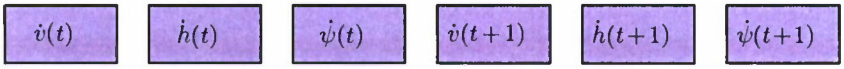
Score = -29393031, Edges = 11, Parameters = 9296



Score = -29465324, Edges = 10, Parameters = 5936



Score = -29791121, Edges = 21, Parameters = 7651840



Score = -407089095, Edges = 0, Parameters = 16

REFERENCES

[1] MITRE, "System safety study of minimum TCAS II," MITRE, Technical Rep. MTR-83W241 (1983).

[2] A. Drumm, "Lincoln Laboratory evaluation of TCAS II Logic Version 6.04a," MIT Lincoln Laboratory, Project Report ATC-240 (1996).

[3] M.P. McLaughlin, "Safety study of the Traffic Alert and Collision Avoidance System (TCAS II)," MITRE Corporation, Technical Rep. MTR 97W32 (1997).

[4] B. Chludzinski, "Lincoln Laboratory evaluation of TCAS II logic version 7," MIT Lincoln Laboratory, Project Report ATC-268 (1999).

[5] T. Arino, K. Carpenter, S. Chabert, H. Hutchinson, T. Miquel, B. Raynaud, K. Rigotti, and E. Vallauri, "Studies on the safety of ACAS II in Europe," Eurocontrol, Technical Rep. ACASA/WP-1.8/210D (2002).

[6] ICAO, "ACAS manual," Eurocontrol, Technical Rep. SCRSP/1-WP/53 (2004).

[7] T.A. Choyce and K.M. Ciaramella, "Test and evaluation of TCAS II logic version 7," Federal Aviation Administration, Technical rep. (2000).

[8] ICAO, "Surveillance, radar and collision avoidance," in *ICAO Standards and Recommended Practices*, vol. IV, annex 10 (1998).

[9] T. Miquel and K. Rigotti, "European encounter model," CENA/Sofréavia and QinetiQ, Technical Rep. ACASA/WP1/186/D (2002).

[10] S. Chabert, "Safety encounter model focused on issue SA01a," CENA/Sofréavia and QinetiQ, Technical Rep. SIRE/WP2/21/D (2005).

[11] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," *Computational Intelligence* 5(3), 142–150 (1989).

[12] K. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of California, Berkeley (2002).

[13] R.D. Grappel, "ASR-9 Processor Augmentation Card (9-PAC) phase II scan-scan correlator algorithms," MIT Lincoln Laboratory, Project Report ATC-298 (2001).

[14] J.L. Gertz, "Mode S surveillance netting," MIT Lincoln Laboratory, Project Report ATC-120 (1983).

[15] F.N. Fritsch and R.E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal of Numerical Analysis* 17(2), 238–246 (1980).

[16] E.M. Shank, "A coordinate conversion algorithm for multisensor data processing," MIT Lincoln Laboratory, Project Report ATC-139 (1986).

[17] W.M. Bolstad, *Introduction to Bayesian Statistics*, Wiley, 2nd ed. (2007).

[18] L.V. Schmidt, *Introduction to aircraft flight dynamics*, Reston, VA: American Institute of Aeronautics and Astronautics (1998).

[19] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection*, Springer (2002).

[20] G.F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning* 9(4), 309–347 (1992).

[21] L.P. Espindle and M.J. Kochenderfer, "Classification of primary radar tracks using Gaussian mixture models," MIT Lincoln Laboratory, In preparation (2008).

[22] J.L. Gertz and R.D. Grappel, "Surveillance improvement algorithms for Airport Surface Detection Equipment Model X (ASDE-X) at Dallas-Fort Worth Airport," MIT Lincoln Laboratory, Project Report ATC-333 (2007).

[23] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, San Francisco, CA: Morgan Kaufmann (1988).

[24] F.V. Jensen, *Bayesian Networks and Decision Graphs*, Springer Verlag (2001).

[25] R.E. Neapolitan, *Learning Bayesian Networks*, Upper Saddle River, NJ: Prentice Hall (2004).

[26] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd ed. (2001).

[27] D. Heckerman, D. Geiger, and D.M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning* 20(3), 197–243 (1995).